

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES
Y AUTOMÁTICA



TRABAJO DE FIN DE GRADO

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

Control Parental: Mi Colegio con Sanders

Parental Control: My School with Sanders

Iván Rodríguez Novoa
Carlos Sanz Tomero

Dirigido por: Matilde Santos Peñas

Curso académico 2020-21

Agradecimientos

Queremos dar las gracias a nuestras directoras, Matilde Santos Peñas y Maria Victoria Lopez Lopez por su ayuda y apoyo, sin su guía este trabajo no habría sido posible. Gracias a nuestros familiares, por su apoyo constante, por brindarnos la oportunidad de crecer y seguir adelante con su esfuerzo constante. A nuestros amigos y compañeros por acompañarnos en esta experiencia universitaria, por esos momentos de desconexión que tanto ayudan a despejar la mente.

Resumen

En la actualidad, los jóvenes comienzan a tener más interiorizado el uso de las redes y las nuevas tecnologías en general, pero en muchas ocasiones no tienen conocimiento de los problemas que puede ocasionar la utilización incorrecta o un abuso del uso de estos medios.

Ejemplos de este problema social se muestran en el informe sobre 'El Estado Mundial de la Infancia', redactado por UNICEF, que recuerda que los menores deben ser protegidos frente a los riesgos que puede ocasionar el uso de Internet, y no porque sea un entorno malo en sí, ya que como se dice en el informe, esa tecnología está en el centro de nuestras vidas. Pero cuanto más se reduzcan los riesgos asociados a su mal uso, más beneficios tendremos derivados de sus ventajas, sobre todo, las nuevas generaciones.

Este Trabajo de Fin de Grado surge de una colaboración con la Fundación Sanders. Es una organización que desarrolla actividades de aprendizaje social y emocional para menores en riesgo de exclusión. Resulta de una acción llevada a cabo para paliar los problemas que surgieron con motivo de la pandemia del Covid-19. Las familias más desfavorecidas no tenían medios para poder desarrollar la educación de sus menores con tecnologías que les permitieran trabajar desde casa. La Fundación Sanders donó herramientas computacionales para algunos de esos niños desfavorecidos, pero como parte de su objetivo educacional, pidieron un seguimiento de las horas dedicadas al uso de estos medios para evitar su uso incorrecto.

En este trabajo se ha desarrollado una aplicación Android que permite a los tutores y profesores monitorizar las horas de uso de los dispositivos tecnológicos cedidos a los estudiantes. Esto les permitirá tomar acciones orientadas a un mejor aprovechamiento de estos recursos. Primero se investigaron distintas herramientas que ya existen y que se usan para monitorizar dispositivos, con el fin de estudiar su modo de operación. A continuación se desarrolla la aplicación móvil. Se describen con detalle el desarrollo, así como todas las herramientas y servicios utilizados, y su implementación. Por último se muestran los resultados, con una secuencia de pasos que describen cómo se utiliza la aplicación y su utilidad.

Los resultados son satisfactorios y es muy gratificante haber contribuido a mejorar la calidad de educación y sobre todo a proteger a los menores de un uso incorrecto de las nuevas tecnologías, y de ayudarles a utilizar bien estos recursos.

Palabras clave: Fundación Sanders, educación digital, control parental, Android Studio, Firebase.

Abstract

At present, young people are beginning to have more internalized the use of networks and new technologies in general, but in many cases they are not aware of the problems that the incorrect use or abuse of the use of these means can cause.

Examples of this social problem are shown in the report on 'The State of the World's Children', written by UNICEF, which recalls that minors must be protected against the risks that the use of the Internet can cause, and not because it is an environment bad in itself, since as stated in the report, that technology is at the center of our lives. But the more the risks associated with its misuse are reduced, the more benefits we will have derived from its advantages, especially the new generations.

This Final Degree Project arises from a collaboration with the Sanders Foundation. It is an organization that develops social and emotional learning activities for minors at risk of exclusion. It is the result of an action carried out to alleviate the problems that arose due to the Covid-19 pandemic. The most disadvantaged families did not have the means to develop the education of their minors with technologies that would allow them to work from home. The Sanders Foundation donated computer tools for some of those underprivileged children, but as part of their educational goal, they asked for hours spent using these media to be tracked to prevent misuse.

In this work, an Android application has been developed that allows tutors and teachers to monitor the hours of use of technological devices given to students. This will allow them to take actions aimed at making better use of these resources. First, different tools that already exist and are used to monitor devices were investigated, in order to study their mode of operation. The mobile application is developed below. The development is described in detail, as well as all the tools and services used, and their implementation. Finally, the results are shown, with a sequence of steps that describe how the application is used and its usefulness.

The results are satisfactory and it is very gratifying to have con-

tributed to improve the quality of education and, above all, to protect minors from misuse of new technologies, and to help them make good use of these resources.

Keywords: Sanders Foundation, digital education, parental control, Android Studio, Firebase.

Índice general

1. Introducción	11
1.1. Motivación	11
1.2. Fundación Sanders	12
1.3. Objetivos Generales	13
1.4. Asignaturas relacionadas y conocimientos adquiridos	13
1.5. Estructura de la Memoria	14
1.6. Motivation	15
1.7. Sanders Foundation	15
1.8. General objectives	16
1.9. Related subjects and acquired knowledge	17
1.10. Report structure	17
2. Estado del Arte	19
2.1. Herramientas de Control Parental	19
2.1.1. Qustodio	19
2.1.2. Family Link	24
2.2. Otras herramientas	31
2.3. Resumen de los datos recogidos	36
3. Desarrollo de la aplicación	39
3.1. Introducción	39
3.2. Organización de roles	39
3.3. Firebase	39
3.4. La aplicación	41
3.4.1. Entorno de desarrollo	41
3.4.2. Estructura del proyecto	41
3.4.3. Integración de Firebase	45
3.4.4. Elementos de Firebase Cloud Firestore	45
3.4.5. Autenticación	50
3.4.6. Menú principal y opciones	51
3.4.7. Listado de webs	58

4. Resultados obtenidos	63
4.1. Autenticación	63
4.2. Inicio y menú de opciones	63
4.3. Lista de webs	65
4.4. Resumen de los resultados	68
5. Conclusiones y Trabajo Futuro	71
5.1. Conclusiones	71
5.2. Trabajo Futuro	72
5.3. Conclusions	73
5.4. Future work	74
6. Aportación de los alumnos	75
6.1. Aportación del alumno Iván Rodríguez Novoa	75
6.2. Aportación del alumno Carlos Sanz Tomero	78

Índice de figuras

1.1. Logo de la Fundación Sanders	13
2.1. Panel de Control de Qustodio	20
2.2. Usuario agregado a Qustodio para su control	21
2.3. Restricciones de acceso de la herramienta Qustodio	21
2.4. Registro de actividad del menor con Qustodio	22
2.5. Restricción de tiempo establecida para el uso	22
2.6. Bloqueo de actividad	23
2.7. Bloqueo por tiempo de uso: tiempo agotado	24
2.8. Panel de miembros de la herramienta Family Link	25
2.9. Vista del controlador de Family Link	26
2.10. Resumen de los controles	27
2.11. Configuración Parental	28
2.12. Permisos de Chrome	29
2.13. Sitios web bloqueados	30
2.14. Acceso web bloqueado	31
2.15. Información de uso	32
2.16. Permisos de aplicaciones	33
2.17. Límite horario	34
2.18. Límite de uso	35
3.1. Diagrama principal con los roles	40
3.2. Diagrama del control web de la aplicación	40
3.3. Firebase Analytics	42
3.4. Proveedores disponibles para Firebase	43
3.5. Cuentas de varios usuarios	44
3.6. Estructura del cloud Firestore	44
3.7. Interfaz de Android Studio	45
3.8. Contenido del archivo AndroidManifest.xml	46
3.9. Directorio de clases	47
3.10. Directorio de interfaces	47
3.11. Apartado Gradle Scripts	48
3.12. Configuración de build.gradle a nivel de proyecto	48

3.13. Configuración de build.gradle a nivel de aplicación	49
3.14. Estructura de datos de un usuario profesor	50
3.15. Estructura de datos de un usuario alumno	51
3.16. Lógica de inicio de sesión	52
3.17. Lógica de registro	52
3.18. Contenido del archivo HomeActivity.java	53
3.19. Contenido del archivo HomeFragment.java	54
3.20. Lógica del registro de uso	55
3.21. Contenido del archivo AlumnoFragment.java	56
3.22. Lógica de consulta de un alumno	57
3.23. Lógica de consulta de un curso	58
3.24. Contenido de la Clase WebList	59
3.25. Método getWebList	60
3.26. Evento de los elementos de la lista	60
3.27. Clase WebBrowser	61
4.1. Pantalla de autenticación	64
4.2. Pantalla de registro	64
4.3. Pantalla de Inicio	65
4.4. Menú de opciones	66
4.5. Alumno Consultado	66
4.6. Menú de opciones	67
4.7. Listado de webs, versión del profesor	68
4.8. Listado de webs, versión del alumno	69
4.9. Contenido de la web cargada	69

Capítulo 1

Introducción

1.1. Motivación

La educación de los alumnos de familias más desfavorecidas socialmente se ha visto seriamente afectada por la pandemia originada por el virus Covid-19. Debido al protocolo de sanidad impuesto para prever los contagios, se restringió la asistencia a los centros educativos. Portátiles, tablets, teléfonos inteligentes, son algunos de los medios que han sido clave para el seguimiento de la educación durante el confinamiento. Sin embargo, no todas las familias han podido disfrutar de estas tecnologías debido a su coste. Por ello algunas organizaciones, como la "Fundación Sanders", han tenido la iniciativa de repartir dispositivos tecnológicos de segunda mano para las familias con menos posibilidades.

Conociendo el uso de los jóvenes de estas nuevas tecnologías, al ver en los alrededores de los colegios cómo entran y salen con la mirada fija en sus móviles, pasando el tiempo en sus redes sociales o chateando con sus amigos, surge la duda de si los dispositivos que la Fundación reparte están siendo usados con la finalidad educativa para la que estaban pensados. Por ello, como exponen en [1] sobre el uso de Internet, es necesaria la existencia del control parental. Aunque la mayoría de aplicaciones requieren una edad mínima para el uso de sus servicios, resulta muy sencillo para estos jóvenes saltarse ese control.

Sin embargo, no se puede pensar como solución cortar este tipo de comportamiento de raíz, sino que se debe enseñar que las nuevas tecnologías no son malas, es más, pueden ser muy beneficiosas si se hace un uso correcto y medido de ellas. El uso abusivo de la red no quiere decir que haya que desconectar totalmente de un mundo que es cada vez más digital, lo que sería incluso perjudicial para su propio desarrollo como personas.

Lo que se pretende con este Trabajo de Fin de Grado es buscar una solu-

ción, colaborando con la Fundación Sanders, para garantizar el uso correcto, y en este caso exclusivamente educativo, de estos dispositivos. No se trata de bloquear su acceso completo a internet, sino permitir que entren en sitios que puedan aportarles conocimientos y seguir con el aprendizaje, y monitorizar las horas de uso para tomar acciones orientadas a un mejor aprovechamiento de estos recursos. Para ello se ha creado una aplicación Android que permite a los tutores conocer el uso que ha hecho cada alumno del dispositivo digital, y a los alumnos reflexionar sobre las horas que dedican a utilizarlos y los fines que buscan.

1.2. Fundación Sanders

La Fundación Sanders[17] es una organización que desarrolla actividades de aprendizaje social y emocional de nueva generación, con tecnologías para mejorar la experiencia educativa en programas extraescolares que ayudan a menores en riesgo de exclusión social y económico. Con motivo la pandemia ocasionada por el Covid-19, la formación a escolares de familias desfavorecidas se complicó al no poder asistir a clases presenciales en los colegios. Para reforzar esta educación se propusieron varios métodos como el apoyo con trabajos extraescolares, manteniendo una comunicación por medio de WhatsApp entre trabajadores y familias, pero no tuvo la eficiencia esperada. Desde la dirección general de educación se consiguió una donación de dispositivos digitales como tabletas, entre otros, por parte de organizaciones que se encargan de dar apoyo a países en vía de desarrollo con las tecnologías.

Una de ellas es Profuturo[15], que tenía financiación suficiente para poder llevar a cabo los propósitos de acceso a la tecnología de muchos estudiantes. Sin embargo, estas tablets tenían un problema. La organización las usaban en los países con una aplicación propia para su uso sin conexión a internet. Pero cuando se quería usar con conexión les impedían o bloqueaban utilidades que necesitaban los estudiantes, como la configuración de límites de acceso a sitios web, descargas de aplicaciones, etc. Podían hacer una instalación de Google Play y configurar el sistema de control parental, pero debían ir dispositivo por dispositivo para configurar el acceso de cada aplicación. Además, el uso gratuito de estas herramientas no estaban preparadas para un número grande de dispositivos, y algunas aplicaciones sólo eran capaces de monitorizar el uso de un solo dispositivo. Al final probaron configurando una cuenta de gmail, cuya contraseña tenían los educadores con acceso a la cuenta, pero era una forma no muy conveniente ni sencilla para implementarla en la práctica. Al probar algunas de las aplicaciones, se vio que eran útiles, pero no era posible adaptarlo a su caso ya que no permitía la flexibilidad necesaria para un buen control del escolar.

Actualmente los dispositivos que están recibiendo son ordenadores de segun-



Figura 1.1: Logo de la Fundación Sanders

da mano, a los cuales se les carga el sistema operativo Maxmadrid[12], un sistema desarrollado con Linux, ya que no hay medios para pagar las licencias de Windows. Además la Comunidad de Madrid tiene una plataforma llamada Educamadrid, que utilizan en los centros educativos, lo cual es una ventaja a la hora de usar ese sistema operativo Maxmadrid, ya que hay una comunidad de profesores que da apoyo en cualquier centro educativo de la Comunidad de Madrid a esa plataforma. Los centros atendidos son centros I+I, a donde derivan a los escolares que tienen un riesgo de exclusión mayor, y los centros de prevención, a los cuales están adscritos familias para prevenir que lleguen a la fase de exclusión.

1.3. Objetivos Generales

El objetivo principal de este trabajo es desarrollar una aplicación Android para ayudar al control y monitorización de los dispositivos de los escolares con el fin de que hagan un uso exclusivamente académico de los mismos y en unos tiempos establecidos.

Para alcanzar este objetivo, se plantearon los siguientes subobjetivos:

- Solucionar el acceso sin control a internet por parte de los alumnos.
- Hacer una investigación de aplicaciones similares ya existentes.
- Desarrollar una interfaz fácil de entender y de navegar por parte de profesores y alumnos.
- Integrar un entorno de base de datos bien documentado y de fácil comprensión.

1.4. Asignaturas relacionadas y conocimientos adquiridos

En este trabajo hemos utilizado conocimientos aprendidos a lo largo del grado como:

- Tecnología de la Programación, ya que la aplicación está desarrollada en Java.
- Bases de datos, porque primero buscamos varias opciones para la gestión de datos hasta que encontramos una que fue la más convincente, además de estructurar un modelo para la gestión de esos datos y la ejecución de consultas en la propia base de datos.
- Diseño de Sistemas Interactivos, porque es donde se aprende a gestionar los proyectos y cómo desarrollar una estructura correcta y sencilla de navegación entre las interfaces creadas.

En este trabajo hemos adquirido conocimientos como:

- Aprender a usar una nueva aplicación, en concreto, Android Studio.
- Gestionar una base de datos NoSQL como Firebase, además de utilizar las herramientas que proporciona.
- Utilizar un sistema de composición de texto como LaTeX para realizar esta memoria.

1.5. Estructura de la Memoria

El resto del documento está organizado de la siguiente forma:

- Capítulo 1: En este capítulo se explica la motivación para realizar el trabajo, así como los objetivos generales y específicos.
- Capítulo 2: Este capítulo contiene la investigación sobre herramientas de control parental que ya existen, así como su análisis y comparación.
- Capítulo 3: En este capítulo se describe el desarrollo de la aplicación realizada, las herramientas utilizadas y su arquitectura.
- Capítulo 4: Se muestran los resultados obtenidos del desarrollo, explicando paso a paso la navegación entre las interfaces así como su funcionalidad.
- Capítulo 5: En este capítulo se presentan las conclusiones y el posible trabajo futuro.
- Capítulo 6: Se exponen las aportaciones de cada uno de los alumnos, explicando cada apartado con los problemas que han ido saliendo y cómo los han solventado.

El código fuente se puede descargar desde el repositorio GitHub:
<https://github.com/TFG-Ivan-Carlos/MiColegioConSanders>

1.6. Motivation

Students' education from the most socially disadvantaged families has been seriously affected by the pandemic caused by the Covid-19 virus. Due to the health protocol imposed to prevent infections, attendance at educational centers was restricted. Laptops, tablets, smartphones, are some of the means that have been key in monitoring education during confinement. However, not all families have been able to enjoy these technologies due to their cost. For this reason, some organizations, such as the "Sanders Foundation", have taken the initiative to distribute second-hand technological devices to families with fewer possibilities.

Knowing the young people's use of these new technologies, when seeing in the surroundings of the schools how they enter and leave with their eyes fixed on their mobile phones, spending time on their social networks or chatting with their friends, the question arises as to whether devices that the Foundation distributes are being used for the educational purpose for which they were intended. For this reason, as stated in [1] about the use of the Internet, the existence of parental control is necessary. Although most applications require a minimum age to use their services, it is very easy for these young people to skip that control.

However, it is not possible to think as a solution to cut off this type of behavior, but it must be taught that new technologies are not bad, indeed, they can be very beneficial if they are used correctly and measured. The abusive use of the Internet does not mean that you have to completely disconnect from a world that is increasingly digital, which would even be detrimental to your own development as people.

The aim of this Final Degree Project is to find a solution, collaborating with the Sanders Foundation, to guarantee the correct use, and in this case exclusively educational, of these devices. It is not about blocking their complete access to the Internet, but rather allowing them to enter sites that can provide them with knowledge and continue learning, and monitor the hours of use to take actions aimed at making better use of these resources. For this, an Android application has been created that allows tutors to know the use that each student has made of the digital device, and for students to reflect on the hours they spend using them and the purposes they seek.

1.7. Sanders Foundation

Sanders Foundation [17] is an organization that develops new generation social and emotional learning activities, with technologies to improve the

educational experience in after-school programs that help minors at risk of social and economic exclusion. Due to the pandemic caused by Covid-19, the training of schoolchildren from disadvantaged families was complicated by not being able to attend face-to-face classes in schools. To reinforce this education, several methods were proposed, such as support with extracurricular jobs, maintaining communication through WhatsApp between workers and families, but it did not have the expected efficiency. The general direction of education obtained a donation of digital devices such as tablets, among others, by organizations that are responsible for supporting developing countries with technologies.

One of them is Profuturo [15], which had sufficient funding to be able to carry out the purposes of access to technology for many students. However, these tablets had a problem. The organization used them in the countries with its own application for use without an internet connection. But when you wanted to use it online, they prevented or blocked utilities that students needed, such as the configuration of access limits to websites, application downloads, etc. They could do a Google Play installation and configure the parental control system, but they had to go device by device to configure access for each application. In addition, the free use of these tools were not prepared for a large number of devices, and some applications were only capable of monitoring the use of a single device. In the end, they tried setting up a gmail account, whose password the educators with access to the account had, but it was not a very convenient or easy way to implement it in practice. When testing some of the applications, it was found that they were useful, but it was not possible to adapt it to his case since it did not allow the necessary flexibility for a good control of the student.

Currently the devices they are receiving are second-hand computers, which are loaded with the Maxmadrid [12] operating system, a system developed with Linux, since there are no means to pay for Windows licenses. In addition, the Community of Madrid has a platform called Educamadrid, which they use in educational centers, which is an advantage when using this Maxmadrid operating system, since there is a community of teachers that provides support in any educational center in the Community. from Madrid to that platform. The centers served are I + I centers, where they refer schoolchildren who have a greater risk of exclusion, and prevention centers, to which families are assigned to prevent them from reaching the exclusion phase.

1.8. General objectives

The main objective of this work is to develop an Android application to help control and monitor the devices of schoolchildren in order to make exclusively academic use of them and at established times.

To achieve this objective, the following sub-objectives were proposed:

- Solve uncontrolled access to the internet by students.
- Do some research on existing similar apps.
- Develop an interface that is easy for teachers and students to understand and navigate.
- Integrate a well-documented and easy-to-understand database environment.

1.9. Related subjects and acquired knowledge

In this work we have used knowledge learned throughout the degree such as:

- Programming Technology, since the application is developed in Java.
- Databases, because we first looked for several options for data management until we found one that was the most convincing, in addition to structuring a model for managing that data and executing queries on the database itself.
- Interactive Systems Design, because it is where you learn to manage projects and how to develop a correct and simple navigation structure between the created interfaces.

In this work we have acquired knowledge such as:

- Learn to use a new application, specifically Android Studio.
- Manage a NoSQL database like Firebase, in addition to using the tools it provides.
- Use a text composition system such as LaTeX to carry out this report.

1.10. Report structure

The rest of the document is organized as follows:

- Chapter 1: This chapter explains the motivation to do the job, as well as the general and specific objectives.
- Chapter 2: This chapter contains research on existing parental control tools, as well as their analysis and comparison.
- Chapter 3: This chapter describes the development of the application made, the tools used and their architecture.

- Chapter 4: The results obtained from the development are shown, explaining step by step the navigation between the interfaces as well as their functionality.
- Chapter 5: This chapter presents the conclusions and possible future work.
- Chapter 6: The contributions of each of the students are exposed, explaining each section with the problems that have appeared and how they have been solved.

Source code can be downloaded from GitHub repository:
<https://github.com/TFG-Ivan-Carlos/MiColegioConSanders>

Capítulo 2

Estado del Arte

Las nuevas tecnologías están a la orden del día. Por ello, el primer paso en todo desarrollo de un trabajo es realizar un análisis del mercado, para comprobar la existencia de herramientas o software desarrollado similares a los que se pretende diseñar.

Como se ha dicho, hoy en día, los jóvenes utilizan las tecnologías digitales de manera habitual, pero no se puede confirmar que hagan un buen uso de ellas. La relación parental y escolar es necesaria para un correcto control de los menores en cuanto su acceso a Internet; además para los padres es complicado afrontar la enseñanza de sus hijos en un medio que no ellos pueden no dominar, en [7] se recoge la experiencia en la sistematización de acciones relacionadas con el papel que cumplen los Centros de Padres y Apoderados en los procesos de búsqueda de una educación de calidad para sus hijos. Existen estudios relacionados con el patrón de uso a la red como en [18] y, sobretodo, de las redes sociales según el rango de edades, donde se investiga el comportamiento y el factor de riesgo de estos jóvenes, y cómo afecta el control parental.

Existen distintas aplicaciones que permiten controlar el buen uso de los dispositivos que proporcionan el acceso a la red. Hay estudios, como [13], que analizan el efecto de estas aplicaciones sobre los jóvenes y el conocimiento de sus mayores sobre ellos. En este apartado investigamos más a fondo lo que estas herramientas pueden proporcionar.

2.1. Herramientas de Control Parental

2.1.1. Qustodio

Qustodio [16] es una aplicación de control parental que permite monitorizar las actividades y accesos de los menores a Internet. Qustodio es una de las primeras aplicaciones que hemos investigado para conocer la metodología que deberíamos usar para crear un software del mismo estilo. Esta

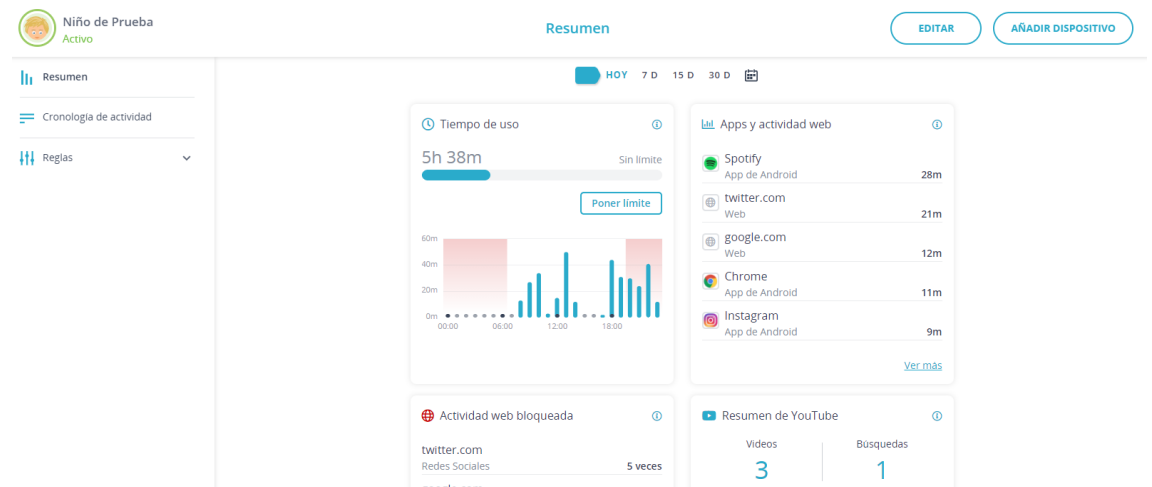


Figura 2.1: Panel de Control de Qustodio

herramienta es compatible con Windows, Mac OS, Android, iOS y kindle. Además sus productos están orientados a familias, escuelas y empresas.

Desarrollo de Qustodio

Primero se ha analizado la versión de prueba gratuita, y pasados 3 días se debe pasar a la versión premium. Una vez instalado el software tanto en nuestro dispositivo como en el que queremos controlar, pasamos a ver el panel de control desde el navegador, como se aprecia en la Figura 2.1. En este panel de control se muestra un resumen donde se recogen todas las actividades que se han realizado en el dispositivo controlado, la cronología de cada acción, y en qué momento se ha realizado. También se pueden observar las reglas, donde se configuran las restricciones del dispositivo. En el apartado de "Tus dispositivos" se pueden agregar los dispositivos de los jóvenes que se quieran controlar. En el caso de la Figura 2.2, se agrega un dispositivo Android de un hijo.

Restricciones

Tras configurar alguna restricción, como el límite de tiempo mostrado en la Figura 2.5, y el acceso que se muestra en la Figura 2.3 a ciertas páginas web, se puede ver toda la actividad (Figura 2.4). Además si en el dispositivo se intenta acceder a un sitio web o aplicación no permitido, salta el control donde se solicita la contraseña de la cuenta (Figura 2.6), la cual el menor bajo control no debería saber.

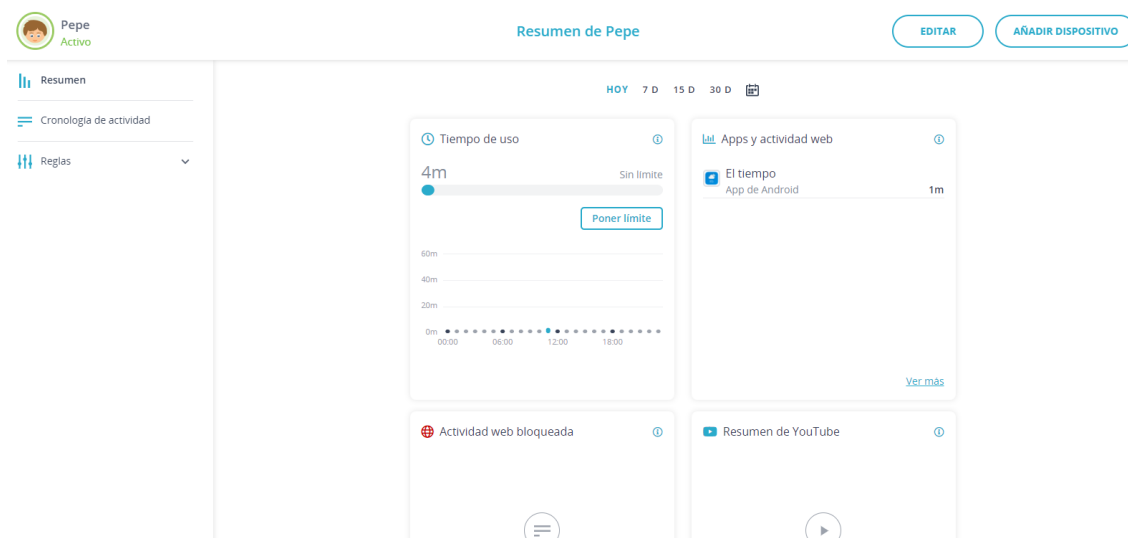


Figura 2.2: Usuario agregado a Qustodio para su control

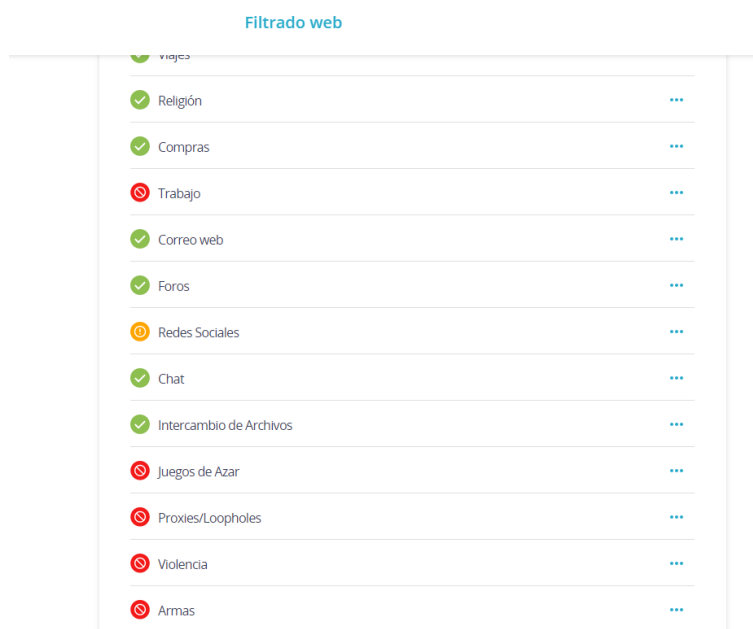


Figura 2.3: Restricciones de acceso de la herramienta Qustodio

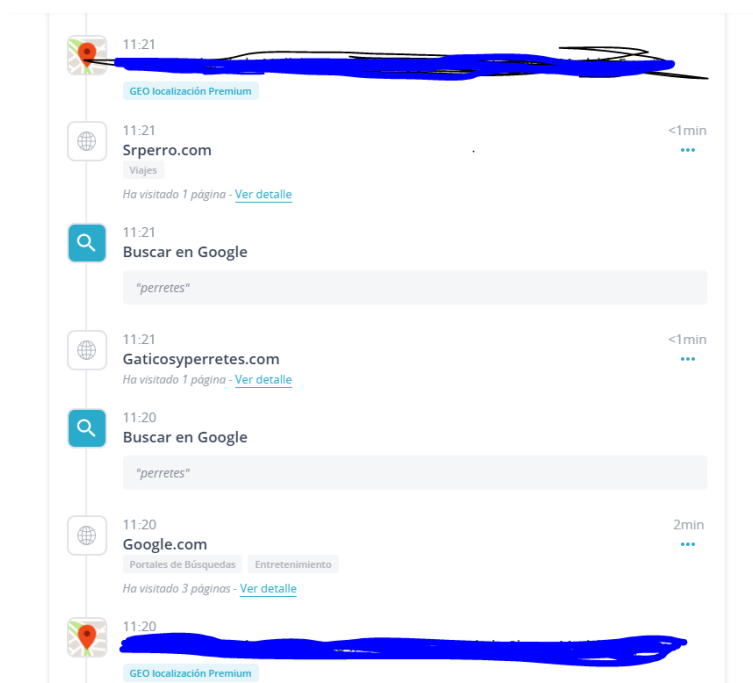


Figura 2.4: Registro de actividad del menor con Qustodio



Figura 2.5: Restricción de tiempo establecida para el uso

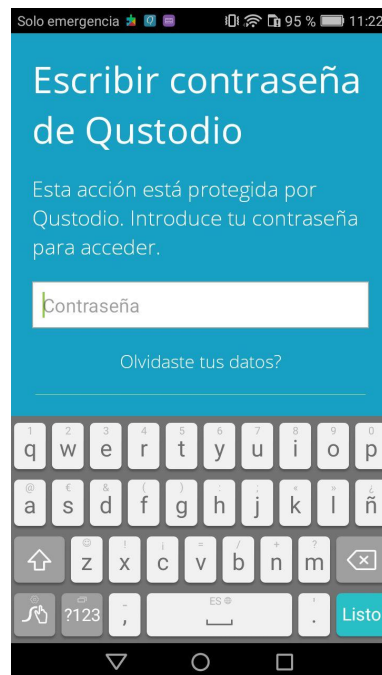


Figura 2.6: Bloqueo de actividad

Tiempo agotado

También se puede definir un tiempo máximo de uso. Una vez agotado este tiempo, el dispositivo no permite acceder a ninguna aplicación, mostrando una pantalla como la de la Figura 2.7.

Ventajas

Los puntos fuertes de esta herramienta son los siguientes:

- Control de acceso de todas las aplicaciones instaladas y sitios web.
- Control de tiempo de uso del dispositivo.
- Notificación de las aplicaciones instaladas vía correo electrónico.
- Visualización de los sitios web visitados, además de la fecha correspondiente.
- Se pueden agregar varios dispositivos.

También presenta algunas desventajas, como:

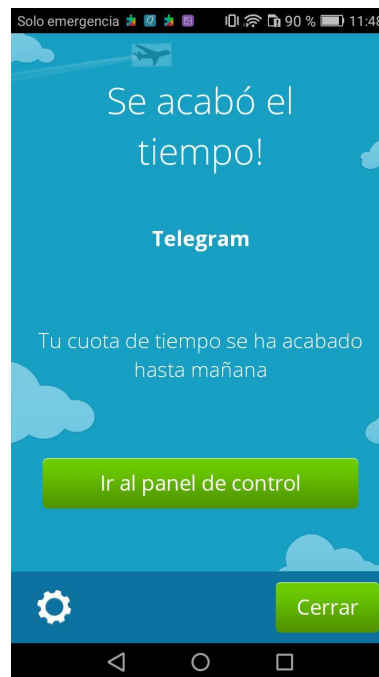


Figura 2.7: Bloqueo por tiempo de uso: tiempo agotado

Desventajas

- Al ser una versión de prueba no se ha podido acceder a todas las funcionalidades.
- El bloqueo no es del todo seguro, ya que si la persona controlada descubre la contraseña puede acceder a cualquier sitio.
- Una vez acabado el tiempo de uso, no bloquea el dispositivo completamente.

2.1.2. Family Link

Family Link [5] es un software desarrollado por Google que permite controlar el uso de un dispositivo con otro. Es una aplicación orientada al control de los dispositivos móviles, que se puede encontrar en la Play Store aunque también se puede configurar desde un ordenador. Se puede gestionar la configuración del dispositivo para realizar todo tipo de control, descargas y compras en Google Play, filtros de búsqueda de Google Chrome, Youtube, ubicación, etc. Por ejemplo, dentro de los filtros de Google Chrome, además de activar una búsqueda segura, podemos gestionar sitios web manualmente. Al hacerlo, si el dispositivo controlado intenta acceder a algún sitio web bloqueado, necesita pedir permiso, en cuyo caso se envía un correo a la cuenta

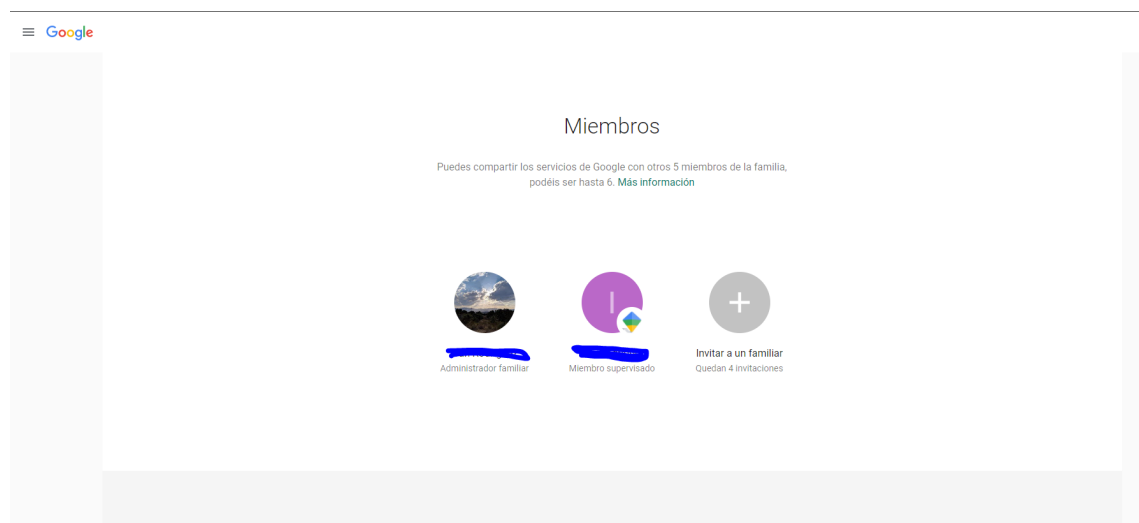


Figura 2.8: Panel de miembros de la herramienta Family Link

de los tutores o mayores encargados, que deciden si permitirlo o no.

Desarrollo de Family Link

Cuando se instala la aplicación en los dispositivos, en la versión de los padres o tutores, pedirá si queremos crear una familia. Lo que realiza es un enlace de cuentas donde la cuenta de Google de los padres es el controlador, y los dispositivos asociados son los hijos. Se pueden agregar varias cuentas a la familia, hasta un límite de 6 miembros. De esta forma, cada aplicación que se instale en el dispositivo controlado lo notificará al correo de los padres. Una vez creada se puede observar el panel de miembros de la familia (Figura 2.8).

Vistas

Desde el dispositivo controlador se observa un panel de control (Figura 2.9) donde se guarda un resumen de todas las actividades y restricciones que se han configurado. Por otra parte, desde el dispositivo controlado solamente se puede observar lo que ven los padres (Figura 2.10). Como se ha dicho, si se quiere realizar alguna acción, necesita pedir permiso para hacerla, es decir, se manda una solicitud al correo electrónico de los padres y estos deciden si dan su permiso.

Configuración

Se puede gestionar la configuración del dispositivo para realizar todo tipo de control, descargas y compras en Google Play, filtros de búsqueda

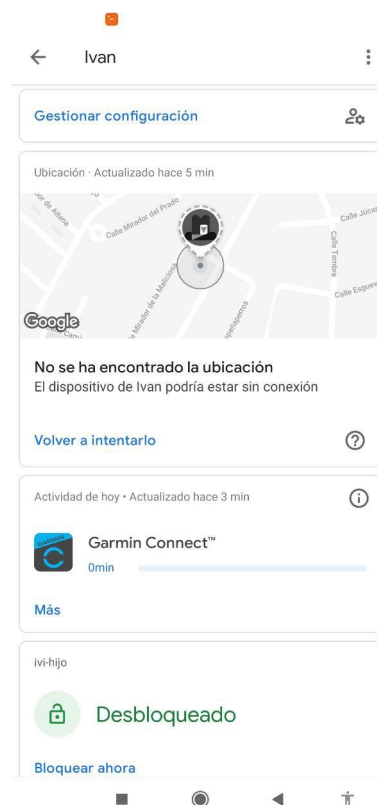


Figura 2.9: Vista del controlador de Family Link

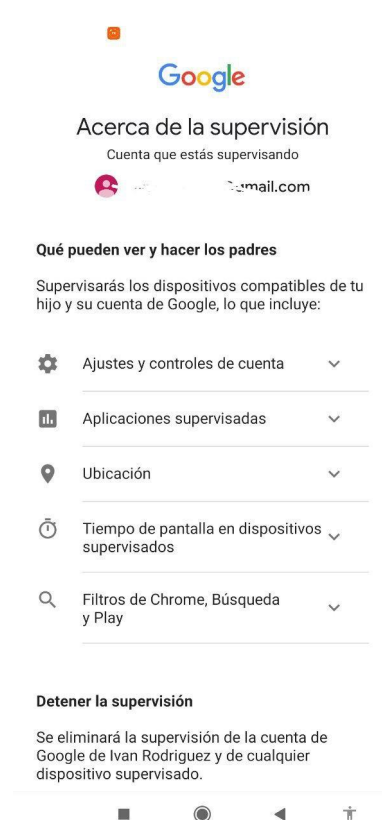


Figura 2.10: Resumen de los controles

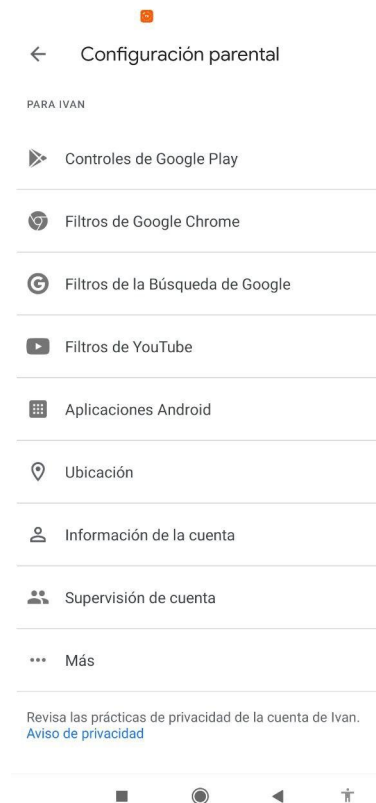


Figura 2.11: Configuración Parental

de Google Chrome, Youtube, ubicación, etc., como muestra la figura 2.11. Por ejemplo, dentro de los filtros de Google Chrome, mostrado en la Figura 2.12, además de activar una búsqueda segura podemos gestionar sitios web manualmente (Figura 2.13). Al hacerlo, si el dispositivo controlado intenta acceder a algún sitio web bloqueado (Figura 2.14), necesita pedir permiso, en cuyo caso se envía un correo a la cuenta de los padres.

Restricciones

También se pueden ver las aplicaciones a las que accede y el tiempo que consume dentro de ellas (Figura 2.15), y se pueden bloquear las aplicaciones que no queremos que sean accedidas (Figura 2.16). Una aplicación bloqueada no aparece en el dispositivo controlado aunque esté instalada. También permite fijar un límite horario (mostrado en la Figura 2.17), además del uso diario del dispositivo (Figura 2.18). Una vez ese tiempo se acaba, el dispositivo pasa a ser bloqueado completamente.

También presenta una serie de ventajas y desventajas, que se pueden resumir en:

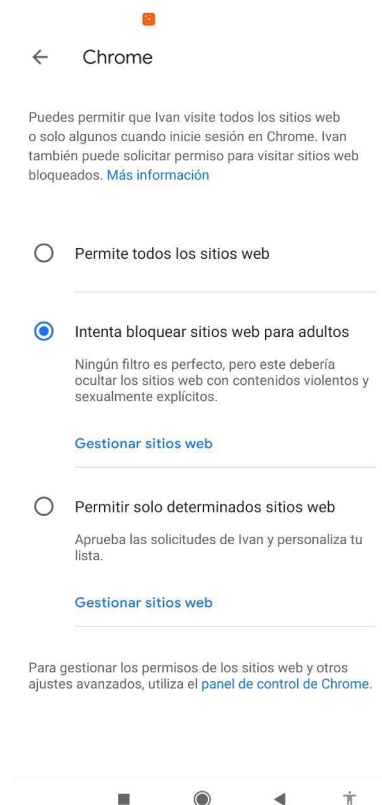


Figura 2.12: Permisos de Chrome

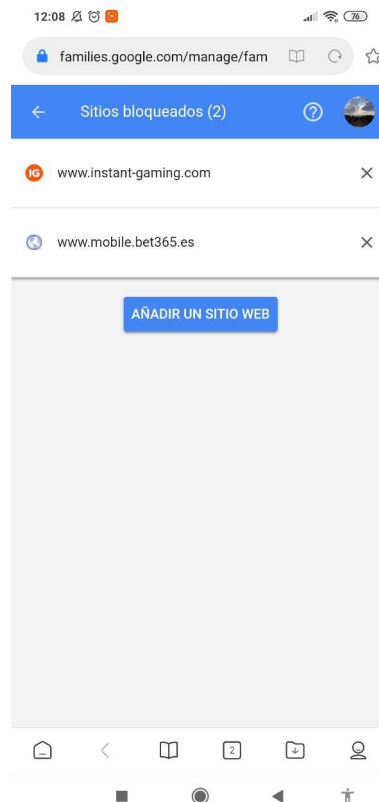


Figura 2.13: Sitios web bloqueados

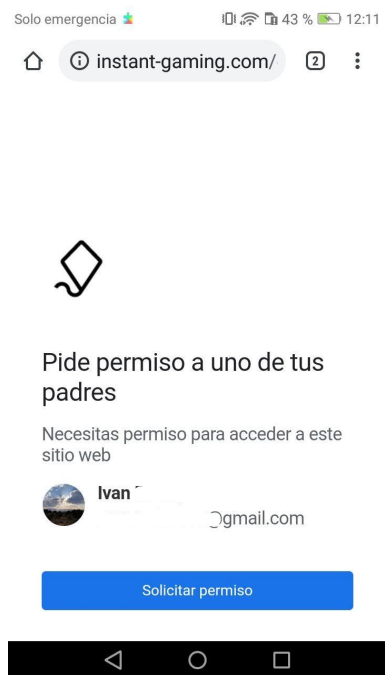


Figura 2.14: Acceso web bloqueado

Ventajas

- Es una aplicación bastante completa e intuitiva.
- Permite de forma segura el uso de los dispositivos de los menores.
- Notifica todas las acciones que realizan los menores.
- Se puede controlar tanto desde un dispositivo Android, como en un ordenador desde la web.
- Es completamente gratuita.

Desventajas

- No muestra los sitios web donde acceden los hijos, solo el tiempo de uso del navegador.

2.2. Otras herramientas

Kroha [14]: es una aplicación de control parental para móviles y tablets que, al igual que Qustodio, permite monitorizar las actividades y accesos de

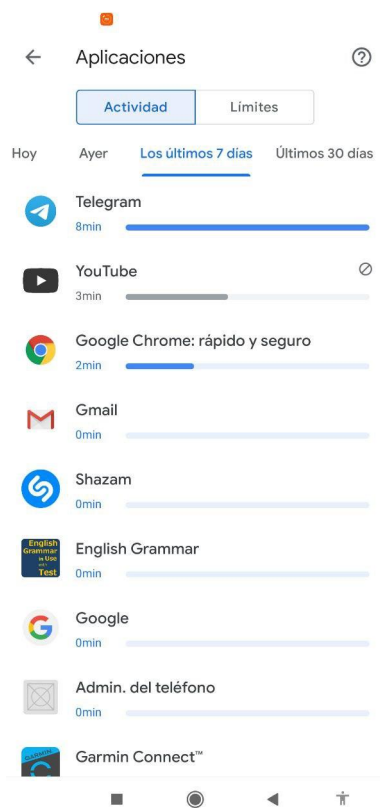


Figura 2.15: Información de uso

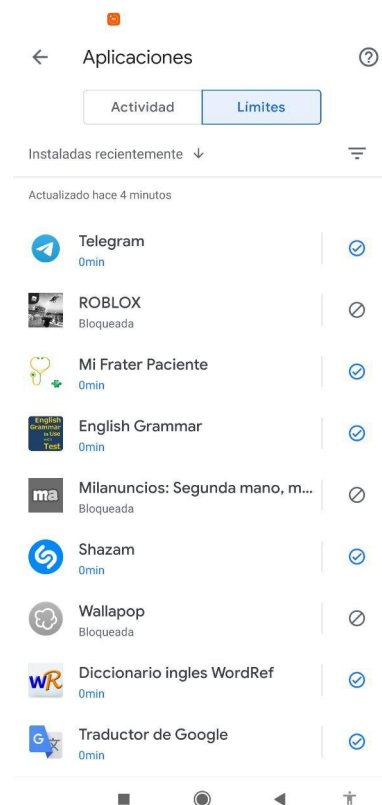


Figura 2.16: Permisos de aplicaciones



Figura 2.17: Límite horario

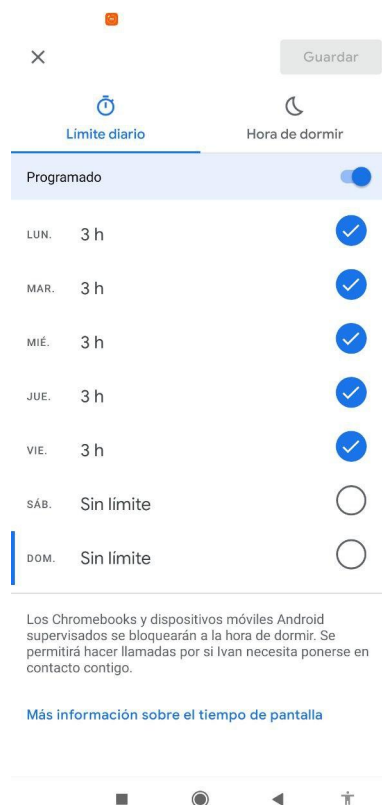


Figura 2.18: Límite de uso

menores a Internet. Este software está únicamente disponible para Google-Play, Apple Store y APPGallery. Se pueden restringir aplicaciones por categorías, así como aplicaciones en concreto, por ejemplo, para que el usuario no pueda instalar cierto tipos de aplicaciones, o sólo pueda usarlas durante cierto tiempo.

Kidbox [3]: KidBox es un programa de control parental y de contenidos para los más pequeños. No sólo ayuda a limitar los lugares a los que pueden ir los niños, o el tiempo de uso y la franja horaria desde la que pueden utilizar el ordenador, sino que también propone una interfaz especialmente adaptada para ellos, de forma que les resulte atractivo el uso o la navegación a través de KidBox. Además presenta contenidos especialmente adaptados para menores, previamente supervisados. KidBox se encuentra todavía en fase beta y su uso es gratuito. Funciona para sistemas Windows SP, Vista y 7. También envían un informe de actividad y podemos saber qué páginas han estado consultando y, en caso de no considerarlas, adecuadas bloquearlas.

Wefisy [9]: es un programa para Windows gratuito que permite a los menores navegar seguros por Internet. Permite bloquear sitios web seleccionando entre más de 30 categorías: pornografía (incluye dominios XXX), phishing, malware, redes sociales, etc., y definir su propia lista blanca y lista negra, determinar horarios de uso, bloquear aplicaciones de P2P, bloquear aplicaciones de mensajería, otras aplicaciones y puertos. Además cuenta con un tutorial en su web para iniciar a los adultos que quieran usar el programa de una forma sencilla.

Kaspersky SafeKids [10]: es otra aplicación para móviles de la compañía Kaspersky Lab. Tiene muchas funcionalidades ya mencionadas en otras aplicaciones. También dispone de filtros para contenidos descargables, pudiendo limitar las descargas de Internet, y notificar su descarga. También permite filtros web tan precisos como búsquedas de información de temas como alcohol, tabaco, contenido para adultos etc.

2.3. Resumen de los datos recogidos

Tras la recogida de datos, mostrada en la Tabla 2.1, la aplicación a desarrollar tendrá todas las características comunes, además de otras funcionalidades como un registro horario de entrada y salida de la aplicación, y un resumen del tiempo que ha usado el dispositivo por día, con el fin de encontrar un modelo de control mejor para corregir el acceso a la red.

Cuadro 2.1: Tabla de recogida de datos

Software	Control web	Limite de apps	Visor web	límite de uso	Localización
Qustodio	✓	✓	✓	✓	✓
Family Link	✓	✓	✗	✓	✓
Kroha	✓	✓	✗	✓	✓
Kidbox	✓	✗	✓	✗	✗
Wefisy	✓	✓	✗	✓	✗
Kaspersky	✓	✓	✗	✓	✓

Capítulo 3

Desarrollo de la aplicación

3.1. Introducción

En este capítulo se describe el desarrollo la aplicación, los esquemas implementados, y la organización de roles. También se indica cómo se quiere utilizar el sistema, y una explicación del gestor de la base de datos que se ha usado, así como la presentación del entorno de desarrollo y la integración de la base de datos con el entorno. Además se exponen algunas de las funcionalidades desarrolladas con fragmentos de código.

Se ha tomado la decisión de desarrollar la aplicación en un sistema Android, y que acepte gran parte de las versiones que existen. El objetivo de ayudar a docentes y estudiantes a usar herramientas de aprendizaje en aplicaciones móviles es lo que nos ha llevado a desarrollar esta herramienta en Android, para que llegue a la máxima cantidad de gente posible.

3.2. Organización de roles

Primero se ha desarrollado un diagrama de clases, utilizando Lucidchart [11], herramienta de diagramación basada en la web, (Figura 3.1), antes de asignar funcionalidades a la aplicación. Por ello cada clase es el rol de una persona en concreto, que puede ser profesor, alumno o administrador. Además, generamos un escenario de uso para uno de los posibles procesos que se podría realizar en la aplicación, reflejado en la Figura 3.2. A modo de ejemplo, en este caso se muestra una de las tareas principales de la herramienta, que es el control web.

3.3. Firebase

En cuanto a la gestión de datos, se ha utilizado Firebase [6], una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada

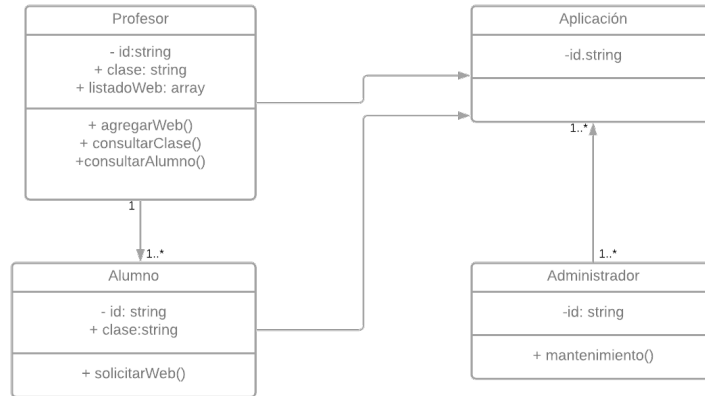


Figura 3.1: Diagrama principal con los roles

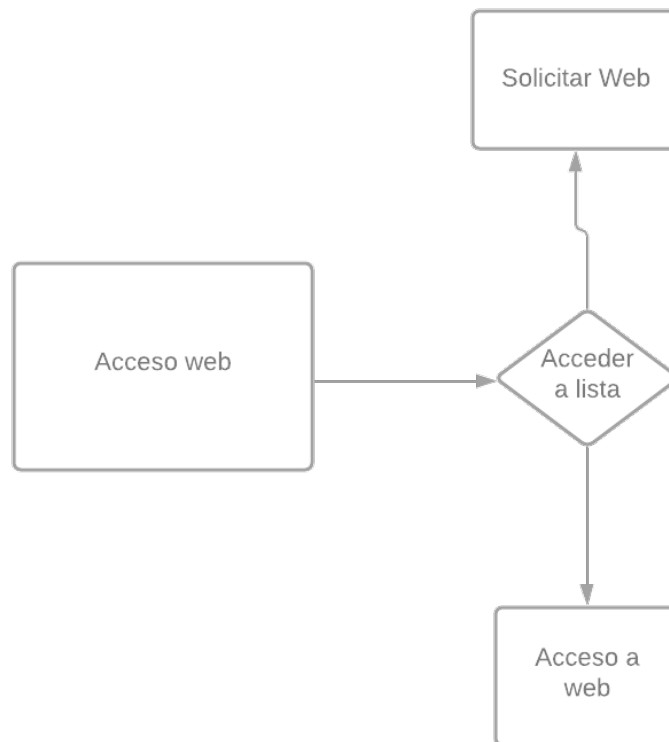


Figura 3.2: Diagrama del control web de la aplicación

por Google. Está ubicada en la nube, integrada con Google Cloud Platform [8]. Usa un conjunto de herramientas para la creación y sincronización de proyectos, con un amplio conjunto de herramientas multiplataforma, y admite variedad de lenguajes de programación. Tiene una documentación muy amplia y accesible con todas las operaciones necesarias para la gestión de los proyectos. En la aplicación a desarrollar se han hecho uso de los siguientes servicios de Firebase:

1. **Firebase Analytics:** Es una aplicación gratuita que proporciona información (también visual) sobre el uso de la aplicación por parte de los usuarios (Figura 3.3).
2. **Firebase Authentication:** Este servicio permite autenticar a los usuarios utilizando únicamente código del lado del cliente. Permite la autenticación con distintos proveedores como Facebook, GitHub, Yahoo, Microsoft, etc. como se muestra en la Figura 3.4 En nuestra aplicación se permite la autenticación por correo electrónico. Creamos varias cuentas de prueba con varios usuarios, como se muestra en la Figura 3.5.
3. **Firebase Cloud Firestore:** Para la gestión de datos de los usuarios de la aplicación, se ha usado el servicio FireStore (Figura 3.6), un servicio de almacenamiento de datos derivado de Google Cloud Platform adaptado para Firebase. Es una base de datos NoSQL[2] que se organiza en forma de documentos agrupados en colecciones, en los que se pueden incluir campos de varios tipos como String, array, boolean, timestamps, objects, entre otros.

3.4. La aplicación

3.4.1. Entorno de desarrollo

Se ha utilizado Android Studio (Figura 3.7), el entorno de desarrollo integrado (IDE) oficial para desarrollar apps para el sistema operativo Android. Admite varios lenguajes de programación. Para esta aplicación se ha decidido usar Java, ya que es el lenguaje con el que más cómodos nos hemos sentido, además de ser uno de los aprendidos a lo largo del grado.

3.4.2. Estructura del proyecto

La estructura de los proyectos de Android Studio está dividida en varias partes. Las más importantes son:

Lo primero que se aprecia es la carpeta manifests/. Todos los proyectos

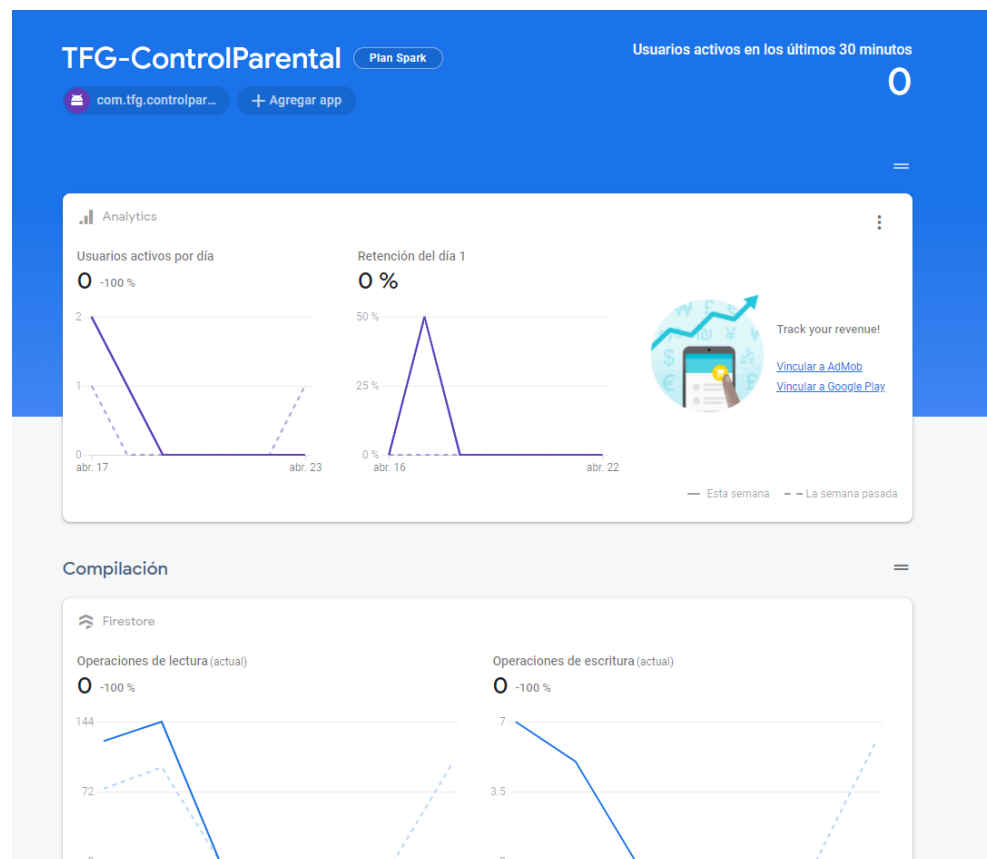
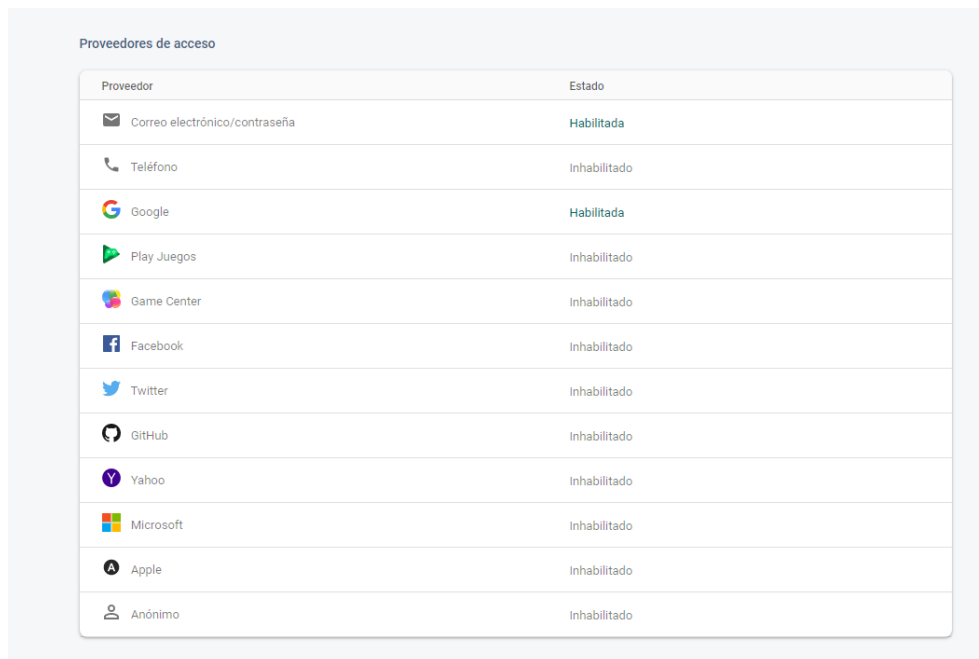


Figura 3.3: Firebase Analytics















Proveedor	Estado
 Correo electrónico/contraseña	Habilitada
 Teléfono	Inhabilitado
 Google	Habilitada
 Play Juegos	Inhabilitado
 Game Center	Inhabilitado
 Facebook	Inhabilitado
 Twitter	Inhabilitado
 GitHub	Inhabilitado
 Yahoo	Inhabilitado
 Microsoft	Inhabilitado
 Apple	Inhabilitado
 Anónimo	Inhabilitado

Figura 3.4: Proveedores disponibles para Firebase

de apps deben tener un archivo `AndroidManifest.xml` (Figura 3.8), con ese mismo nombre, en la raíz de la fuente del proyecto. Este archivo describe información esencial de la aplicación para las herramientas de creación de Android, el nombre del paquete de la aplicación, los componentes, que son las actividades, servicios, receptores de emisiones y proveedores de contenido, y los permisos necesarios para acceder a las partes protegidas del sistema.

Después se encuentra el directorio `java/` (Figura 3.9), en el que se alojan todos los archivos `.class` que componen la funcionalidad de la aplicación. La clase `Activity` es un componente clave de una app para Android, así como la forma en que se inician y se crean las actividades. El sistema Android inicia el código en una instancia de `Activity` invocando métodos de devolución de llamada específicos que corresponden a etapas específicas de su ciclo de vida.

En el directorio `res/` (Figura 3.10) se agregan los recursos de la aplicación. En este caso solamente se han agregado archivos de diseño, pero se pueden incluir mapas de bits, archivos SVG, de string, etc., entre otros.

Por último, el apartado de Gradle Scripts (Figura 3.11), es básicamente un paquete de herramientas avanzadas de compilación para automatizar y administrar el proceso de compilación y, al mismo tiempo, definir configuraciones de compilación personalizadas y flexibles.

✦ Crea prototipos y haz pruebas de extremo a extremo con Local Emulator Suite, que ahora es compatible con Firebase Authentication. [Comenzar](#)

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario ↑
ivan.rodnov13@gmail.com	✉	6 abr. 2021	6 abr. 2021	6kMyxuY6juUwExJD3NJfnhSjqgV2
fsefsdfsdfsdfs@gmail.com	✉	7 abr. 2021	7 abr. 2021	CaxlymNhJwXncliwU7CX4GLh8W...
miguel@gmail.com	✉	18 abr. 2021	19 abr. 2021	Hggsq5EubIO8AToyin2zfSfyXHW2
pruebaok@gmail.com	✉	7 abr. 2021	7 abr. 2021	Oq9GhS6S0HPLZ42YEx6kcxXlwY...
ivan.rodnov@gmail.com	✉	20 mar. 2021	19 abr. 2021	V71WktJUcTud5sqspGfz3lqmdSf1
dsadadafgddad@gmail.com	✉	7 abr. 2021	7 abr. 2021	iILvtfWH9sV89xPoYV8KulZbO0n2
dsadadadad@gmail.com	✉	7 abr. 2021	7 abr. 2021	mZIsjXluNzdWUGVPQ4PGq3utxpG3
ertergderge@gmail.com	✉	7 abr. 2021	7 abr. 2021	oh4JWtkeQ0ZSRXtN6OMdCCoAZ...
tfgcontrolparental@gmail.co...	✉	20 mar. 2021		sTA5fcFIMPbYpvcWf9L3WWjhS4h2

Filas por página: 50 1 - 9 of 9

Figura 3.5: Cuentas de varios usuarios

Cloud Firestore

Datos Reglas Índices Uso

✦ Crea prototipos y haz pruebas de extremo a extremo con Local Emulator Suite, que ahora es compatible con Firebase Authentication. [Comenzar](#)

users	users	miguel@gmail.com
+ Iniciar colección users	+ Agregar documento W9HY4kfn8ha5Qc6J8QD dsadadafgddad@gmail.com fsefsdfsdfsdfs@gmail.com ivan.rodnov@gmail.com miguel@gmail.com pruebaok@gmail.com	+ Iniciar colección + Agregar campo Clase: "3A" Colegio: "Los Valles" Hora_in: 19 de abril de 2021 a las 17:13:50 UTC+2 Hora_out: 19 de abril de 2021 a las 17:14:20 UTC+2 Tutor: "ivan.rodnov@gmail.com" id: "007yu" registro_uso Lunes Horas: 0 Min: 1 Secs: 30' rol: "alumno"

Figura 3.6: Estructura del cloud Firestore

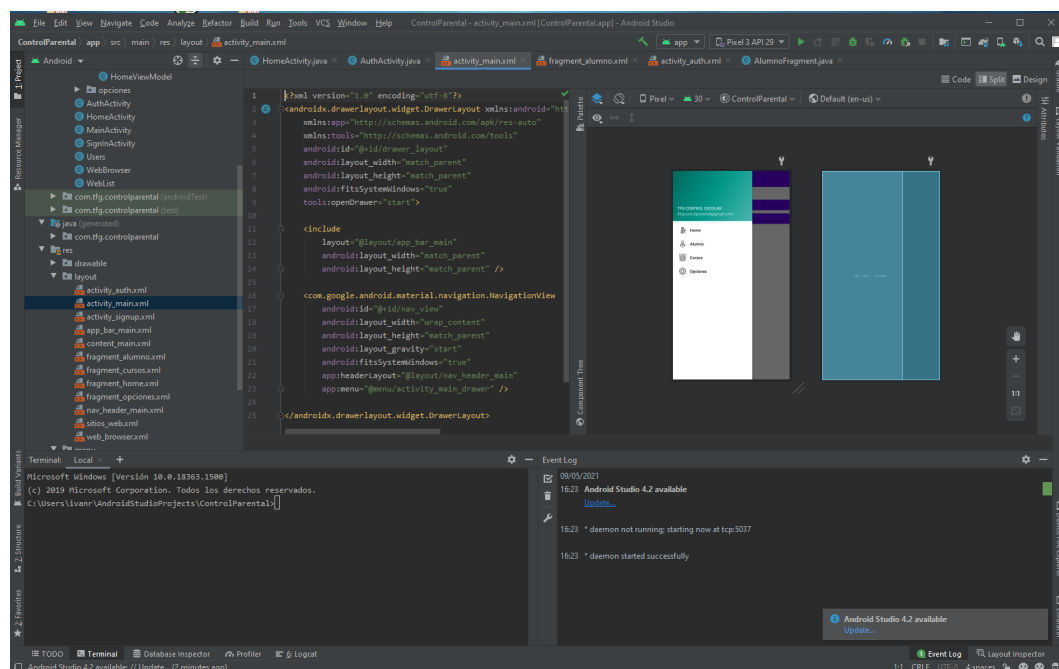


Figura 3.7: Interfaz de Android Studio

3.4.3. Integración de Firebase

Lo primero que se realizó fue conectar el servicio de Firebase al proyecto de la aplicación. Para ello se necesita configurar el archivo build.gradle tanto a nivel de proyecto (Figura 3.12), como de aplicación (Figura 3.13). Para el archivo a nivel de proyecto basta con agregar el repositorio de Google para cargar sus servicios. En cuanto a nivel de aplicación, se necesita agregar las distintas dependencias de los servicios de Firebase que vayan a usarse. Todo ello está muy bien explicado en el apartado de documentación de Firebase [4].

3.4.4. Elementos de Firebase Cloud Firestore

Para la gestión de los usuarios de la aplicación hay una única colección llamada Users. Firestore permite agregar documentos a las colecciones y, a su vez, dentro de estos documentos es posible agregar más colecciones y campos, creando así una estructura de árbol de datos. En este caso, dentro de la colección Users están los identificadores de los usuarios. El registro de estos es mediante un correo electrónico, que les identificará, y así se controla que los usuarios no se repitan o intenten registrarse varias veces. Dentro de un documento existen varios campos. El campo rol es uno de los más importantes, ya que habrá controles dentro de la aplicación según el rol del usuario, que puede ser profesor o alumno. Ambos tipos de usuarios tienen

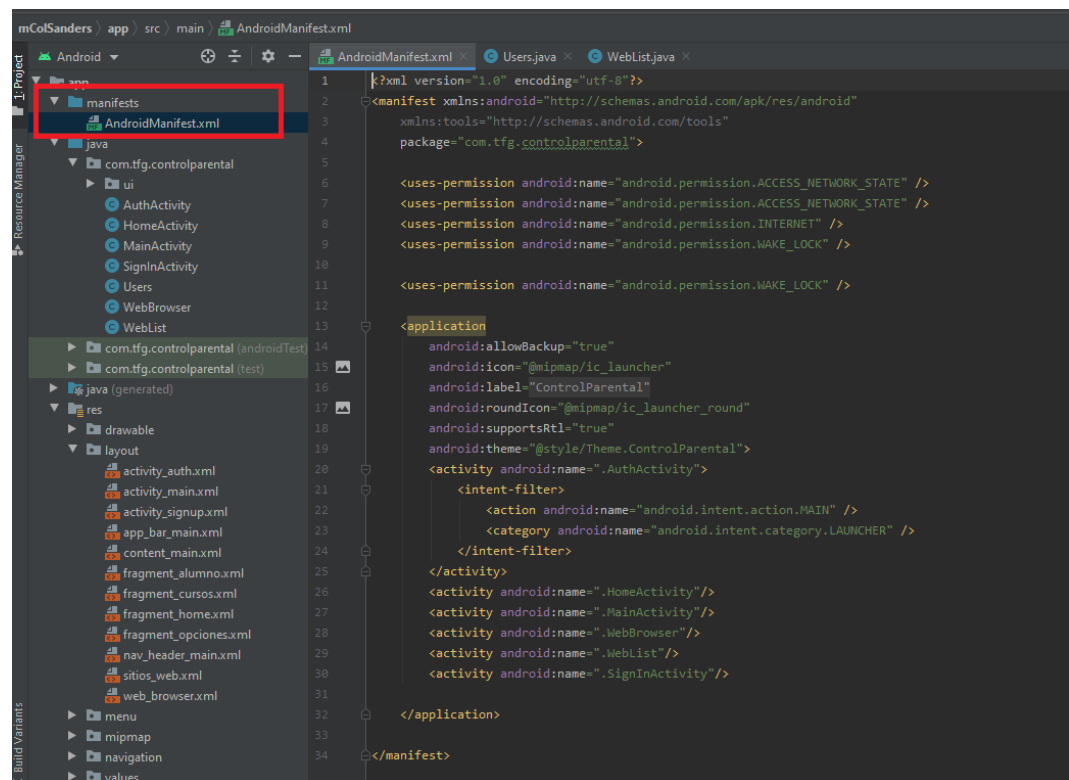


Figura 3.8: Contenido del archivo AndroidManifest.xml

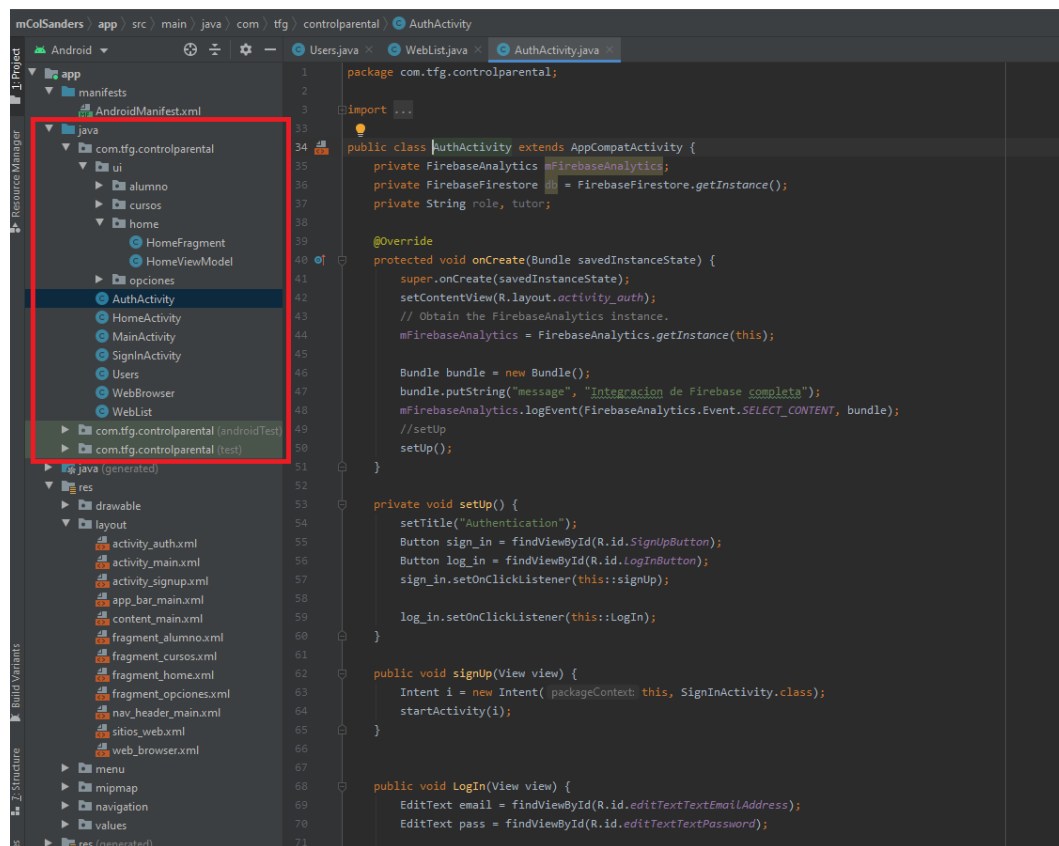


Figura 3.9: Directorio de clases

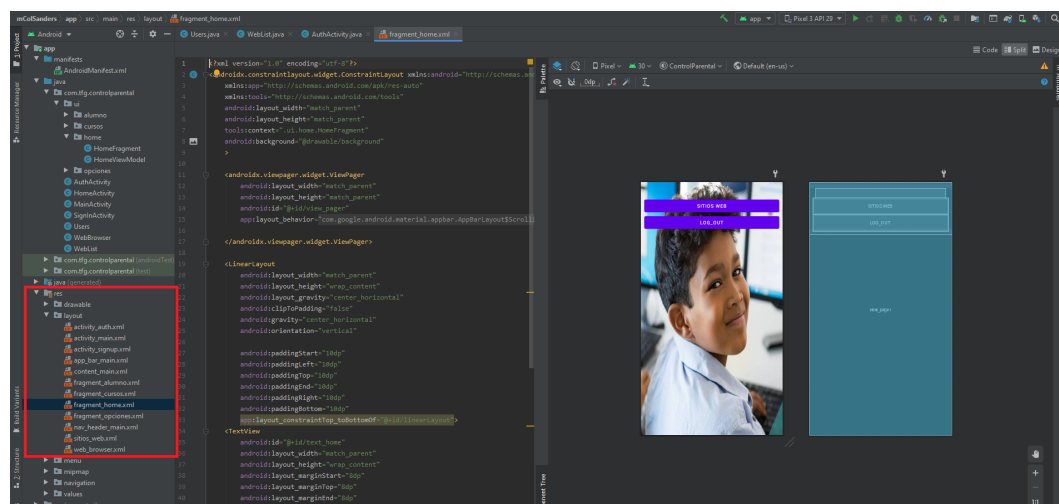


Figura 3.10: Directorio de interfaces

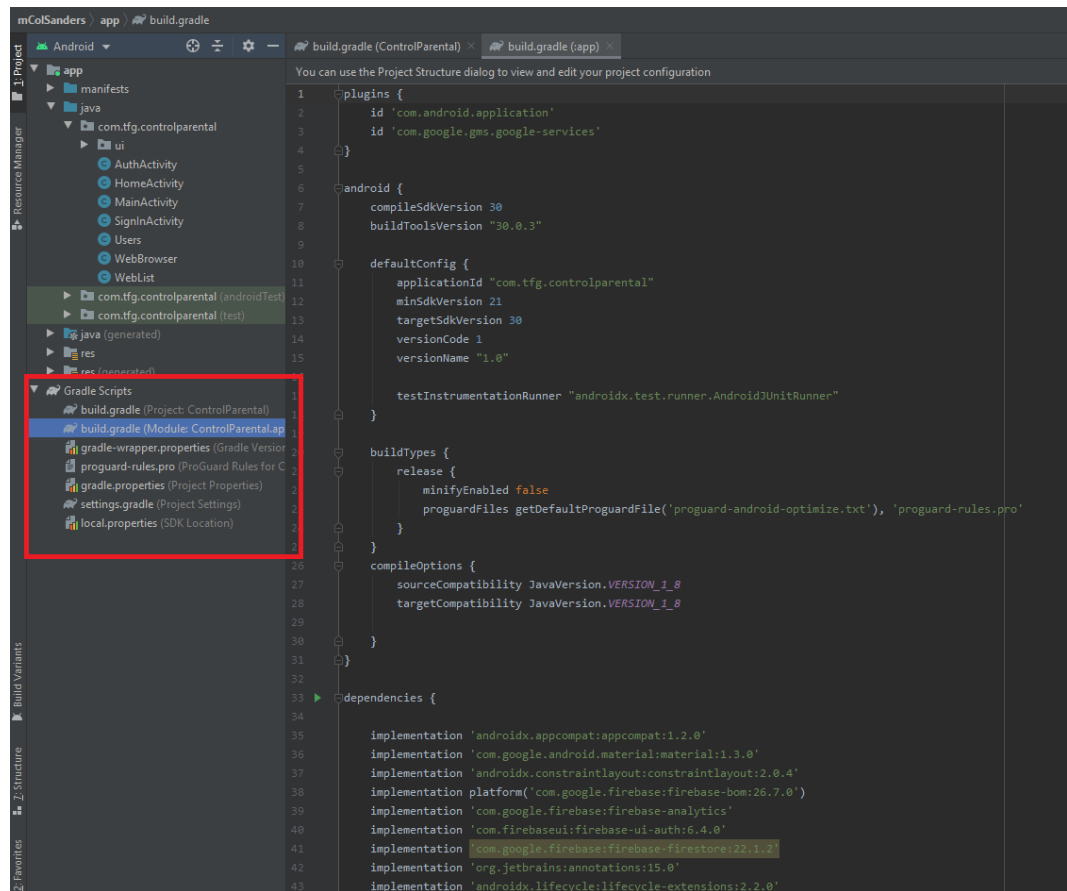


Figura 3.11: Apartado Gradle Scripts

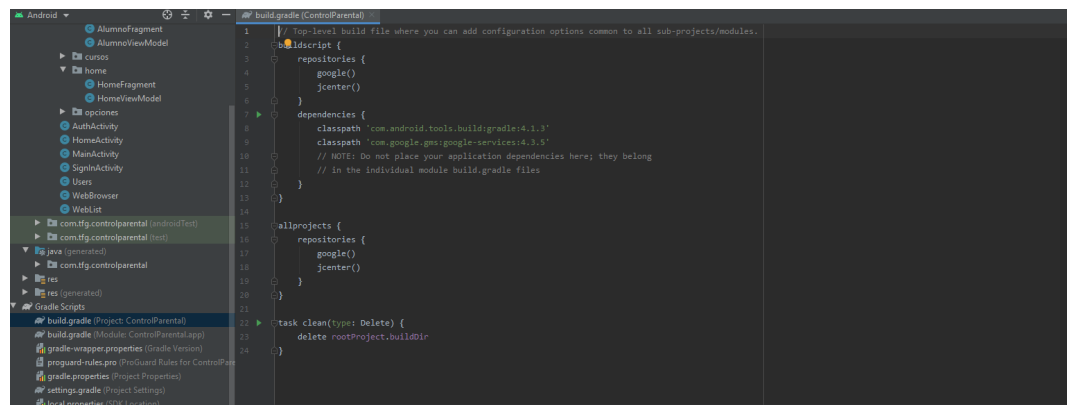


Figura 3.12: Configuración de build.gradle a nivel de proyecto

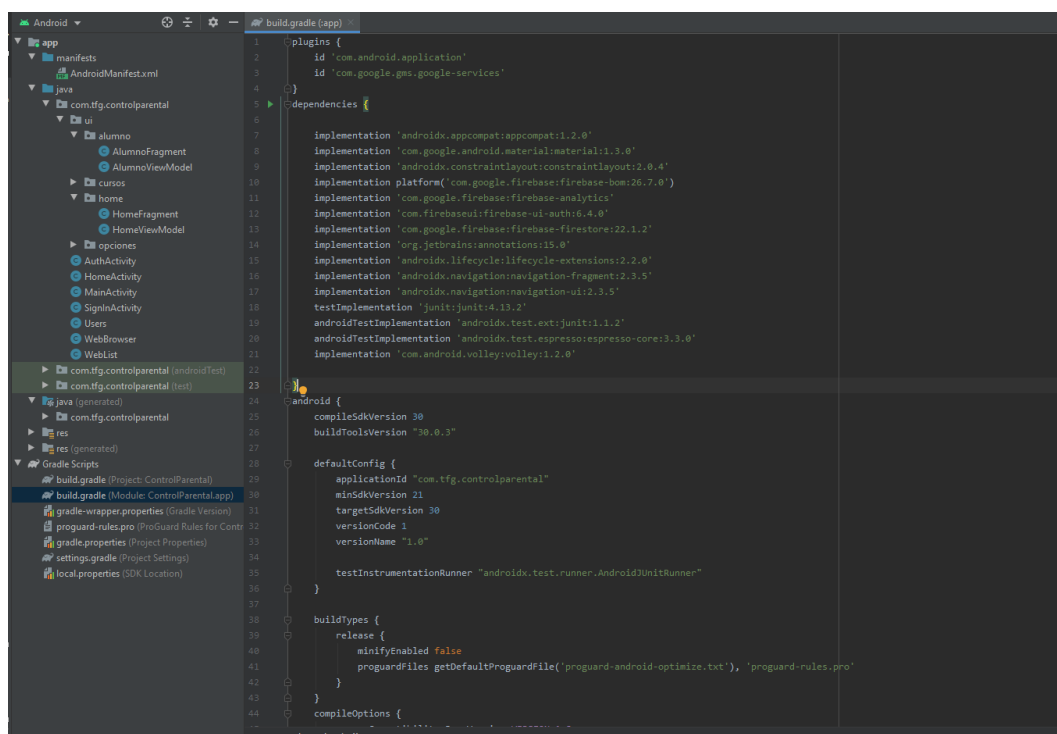


Figura 3.13: Configuración de build.gradle a nivel de aplicación

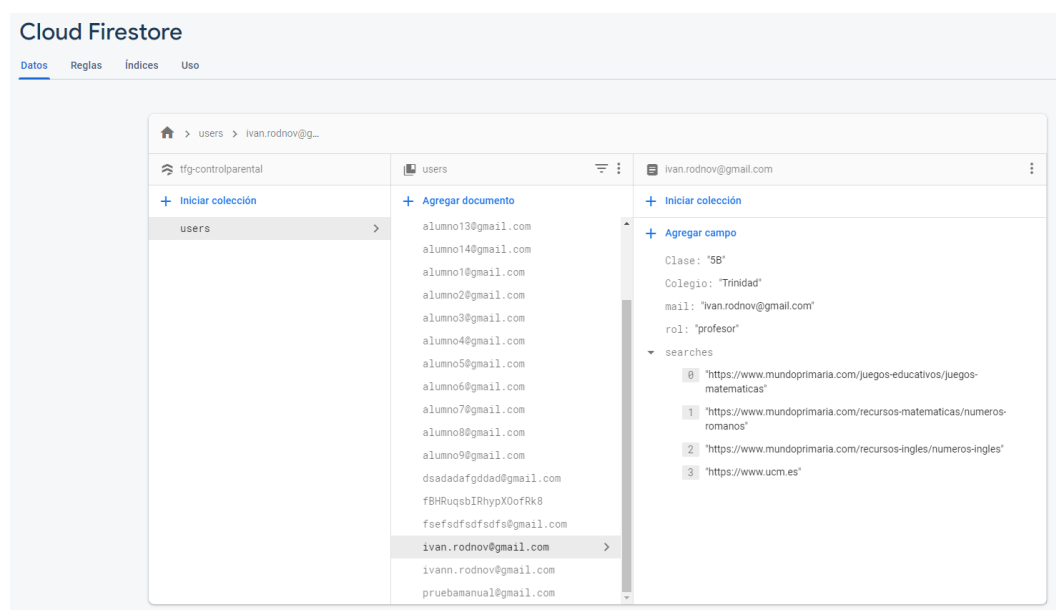


Figura 3.14: Estructura de datos de un usuario profesor

campos en común, como el colegio y la clase a la que pertenecen. El usuario con rol de profesor (Figura 3.14) contiene un campo de control de uso de la aplicación, para que los alumnos de los que sea su tutor no pasen del límite de horas asignado. Como el control de la red es algo muy complejo y fácil de evitar, se decidió que cada profesor tuviera una lista de webs, revisadas con anterioridad por parte del profesor, permitidas. De esta manera los alumnos únicamente tendrán acceso a las webs que su tutor haya aceptado. El usuario con rol de alumno (Figura 3.15), tiene un campo característico: el registro de horas de uso de la aplicación. Este registro es calculado mediante la fecha y hora en la que el alumno ha iniciado sesión y el momento en que cierra sesión.

3.4.5. Autenticación

El primer servicio desarrollado fue la lógica de autenticación gestionada por Firebase Authentication. Está formado por dos clases Java, `AuthActivity.java` (Figura 3.16), con el proceso de inicio de sesión, y `RegActivity.java` (Figura 3.17), donde se aloja la lógica del registro.

Entendemos que el inicio de sesión se realiza para un usuario que haya sido registrado previamente. Para este proceso basta con llamar a una función de `FirebaseAuth` llamada `signInWithEmailAndPassword()`. Esta llamada se encarga de hacer las comprobaciones necesarias para que un usuario que no esté registrado no acceda al sistema. En esta llamada agregamos un evento

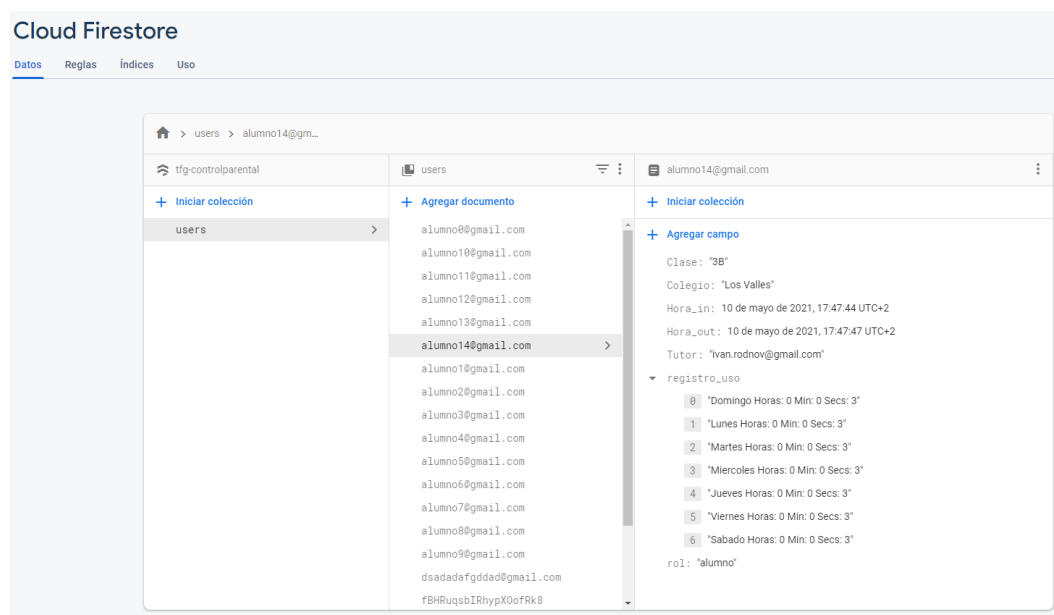


Figura 3.15: Estructura de datos de un usuario alumno

de finalización, `addOnCompleteListener`, que a su vez llama a un método, `onComplete()`, donde hacemos las comprobaciones de si ha resultado exitosa la operación. Si ha habido algún fallo se muestra una alerta; en caso contrario la aplicación redirige a la pantalla del menú principal.

La funcionalidad del registro sigue la misma estructura que la del inicio de sesión, únicamente cambia la llamada a la función, la cual es `createUserWithEmailAndPassword()`, que realiza las comprobaciones de que ese usuario ya exista en el sistema. Para redirigir a otra pantalla se usa un `Intent`, un objeto que proporciona vinculación en tiempo de ejecución entre componentes separados, como dos actividades, en este caso para iniciar otra actividad.

3.4.6. Menú principal y opciones

El menú principal está formado por una clase Java, `HomeActivity.java`, y varios fragmentos reunidos en un directorio `ui/`. La clase `HomeActivity.java` (Figura 3.18) se encarga de la configuración de la barra de navegación donde se definen los `Fragments`. Un `Fragment` representa un comportamiento o una parte de la interfaz de usuario en una `FragmentActivity`. Pueden combinarse varios fragmentos en una sola actividad para crear una interfaz multipanel, y volver a usar un fragmento en diferentes actividades. Se puede pensar en un fragmento como una sección modular de una actividad que tiene un ciclo de vida propio, que recibe sus propios eventos de entrada, y que se puede agregar o quitar mientras la actividad se esté ejecutando. En `HomeFragment`

```

AuthActivity.java
63 Intent i = new Intent( packageContext, this, SignInActivity.class);
64 startActivity(i);
65 }
66
67
68 public void login(View view) {
69     EditText email = findViewById(R.id.editTextEmailAddress);
70     EditText pass = findViewById(R.id.editTextPassword);
71
72     if(email.getText().toString().isEmpty() || pass.getText().toString().isEmpty()){
73         FirebaseAuth.getInstance().signInWithEmailAndPassword(email.getText().toString(), pass.getText().toString()).addOnCompleteListener( activity, this, new OnCompleteListener<AuthResult>() {
74
75             @Override
76             public void onComplete(@NonNull Task<AuthResult> task) {
77                 if(task.isSuccessful()){
78                     // Sign in success, update UI with the signed-in user's information
79                     showHome(email.getText().toString());
80                 }else{
81                     // If sign in fails, display a message to the user.
82                     showAlert();
83                 }
84             }
85         });
86     }
87 }
88
89 //funcion que muestra un pop up al dar error e inicio de sesion
90 private void showAlert() {
91     AlertDialog.Builder alert = new AlertDialog.Builder( context, this);
92     alert.setMessage("Error al autenticar usuario");
93     alert.setPositiveButton( text: "Aceptar", listener: null);
94     AlertDialog pop = alert.create();
95     alert.show();
96 }
97
98 //funcion que pasa a la pantalla Home
99 private void showHome(String user){
100     Intent home = new Intent( packageContext, this, HomeActivity.class);
101     home.putExtra( "name", user);
102     startActivity(home);
103 }
104 }

```

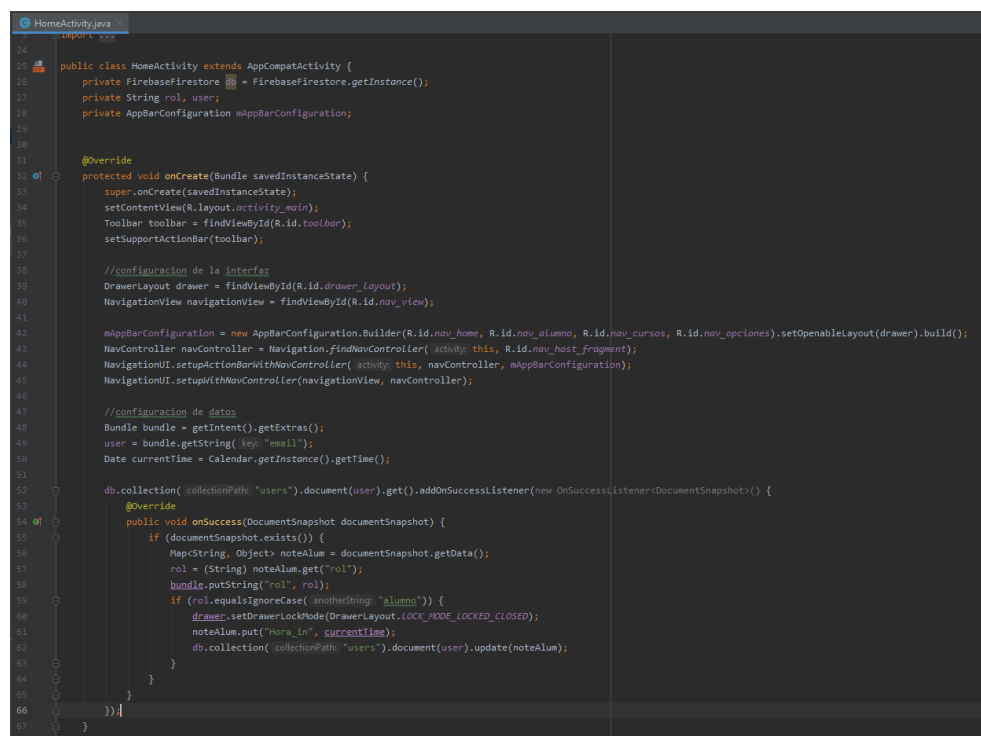
Figura 3.16: Lógica de inicio de sesión

```

RegActivity.java
29 private FirebaseFirestore db = FirebaseFirestore.getInstance();
30 private EditText class, colegio, pass, email;
31
32 @Override
33 protected void onCreate(Bundle savedInstanceState) {
34     super.onCreate(savedInstanceState);
35     setContentView(R.layout.activity_signup);
36
37     //setup
38     setup();
39 }
40
41 private void setup() {
42     setTitle("Registration");
43
44     email = findViewById(R.id.editMail);
45     class = findViewById(R.id.textClass);
46     colegio = findViewById(R.id.textColegio);
47     pass = findViewById(R.id.editPass);
48     Button sign_in = findViewById(R.id.btnRegistro);
49
50     sign_in.setOnClickListener(new View.OnClickListener() {
51         @Override
52         public void onClick(View v) {
53             if(email.getText().toString().isEmpty() || pass.getText().toString().isEmpty() || class.getText().toString().isEmpty() || colegio.getText().toString().isEmpty()){
54                 FirebaseAuth.getInstance().createUserWithEmailAndPassword(email.getText().toString(), pass.getText().toString()).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
55
56                     @Override
57                     public void onComplete(@NonNull Task<AuthResult> task) {
58                         if(task.isSuccessful()){
59                             // Sign in success, update UI with the signed-in user's information
60                             showHome(email.getText().toString());
61                         }else{
62                             // If sign in fails, display a message to the user.
63                             showAlert();
64                         }
65                     }
66                 });
67             }else{
68                 showAlert();
69             }
70         }
71     });
72 }

```

Figura 3.17: Lógica de registro



```

24
25 public class HomeActivity extends AppCompatActivity {
26     private FirebaseFirestore db = FirebaseFirestore.getInstance();
27     private String rol, user;
28     private AppBarConfiguration mAppBarConfiguration;
29
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_main);
35         Toolbar toolbar = findViewById(R.id.toolbar);
36         setSupportActionBar(toolbar);
37
38         //configuracion de la interfaz
39         DrawerLayout drawer = findViewById(R.id.drawer_layout);
40         NavigationView navigationView = findViewById(R.id.nav_view);
41
42         mAppBarConfiguration = new AppBarConfiguration.Builder(R.id.nav_home, R.id.nav_alumno, R.id.nav_cursos, R.id.nav_opciones).setOpenableLayout(drawer).build();
43         NavController navController = Navigation.findNavController( activity, this, R.id.nav_host_fragment);
44         NavigationUI.setupActionBarWithNavController( activity, this, navController, mAppBarConfiguration);
45         NavigationUI.setupWithNavController(navigationView, navController);
46
47         //configuracion de datos
48         Bundle bundle = getIntent().getExtras();
49         user = bundle.getString( key "email");
50         Date currentTime = Calendar.getInstance().getTime();
51
52         db.collection( collectionPath: "users").document(user).get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
53             @Override
54             public void onSuccess(DocumentSnapshot documentSnapshot) {
55                 if (documentSnapshot.exists()) {
56                     Map<String, Object> noteAlum = documentSnapshot.getData();
57                     rol = (String) noteAlum.get("rol");
58                     bundle.putString("rol", rol);
59                     if (rol.equalsIgnoreCase( anotherString: "alumno")) {
60                         drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);
61                         noteAlum.put("Hora_In", currentTime);
62                         db.collection( collectionPath: "users").document(user).update(noteAlum);
63                     }
64                 }
65             }
66         })
67     }

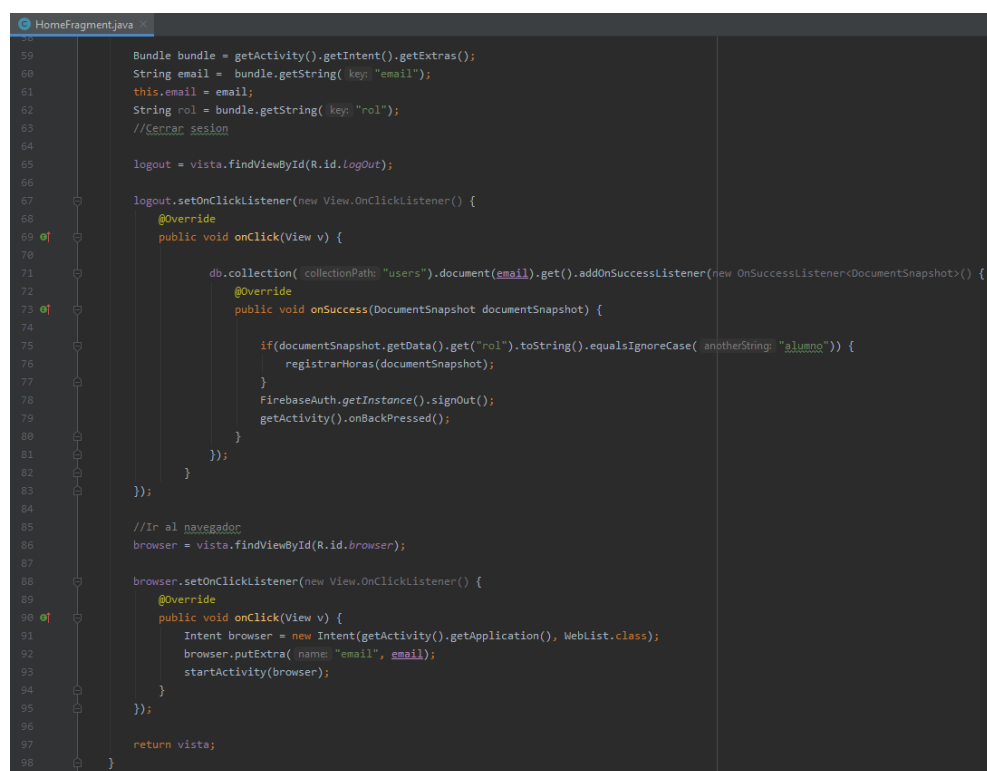
```

Figura 3.18: Contenido del archivo HomeActivity.java

(Figura 3.19) están las funciones de cerrar sesión, donde además se encuentra la lógica encargada del registro horario (Figura 3.20), y el acceso a la lista de webs. Se llama a un Intent para que cargue la actividad WebList explicada más adelante, que básicamente contiene los elementos de la pantalla principal. El acceso a los demás fragmentos es único para un usuario con el rol de profesor, ya que se recogen funciones de consultas a la base de datos, a las cuales no debe tener acceso un alumno. Estas funcionalidades son:

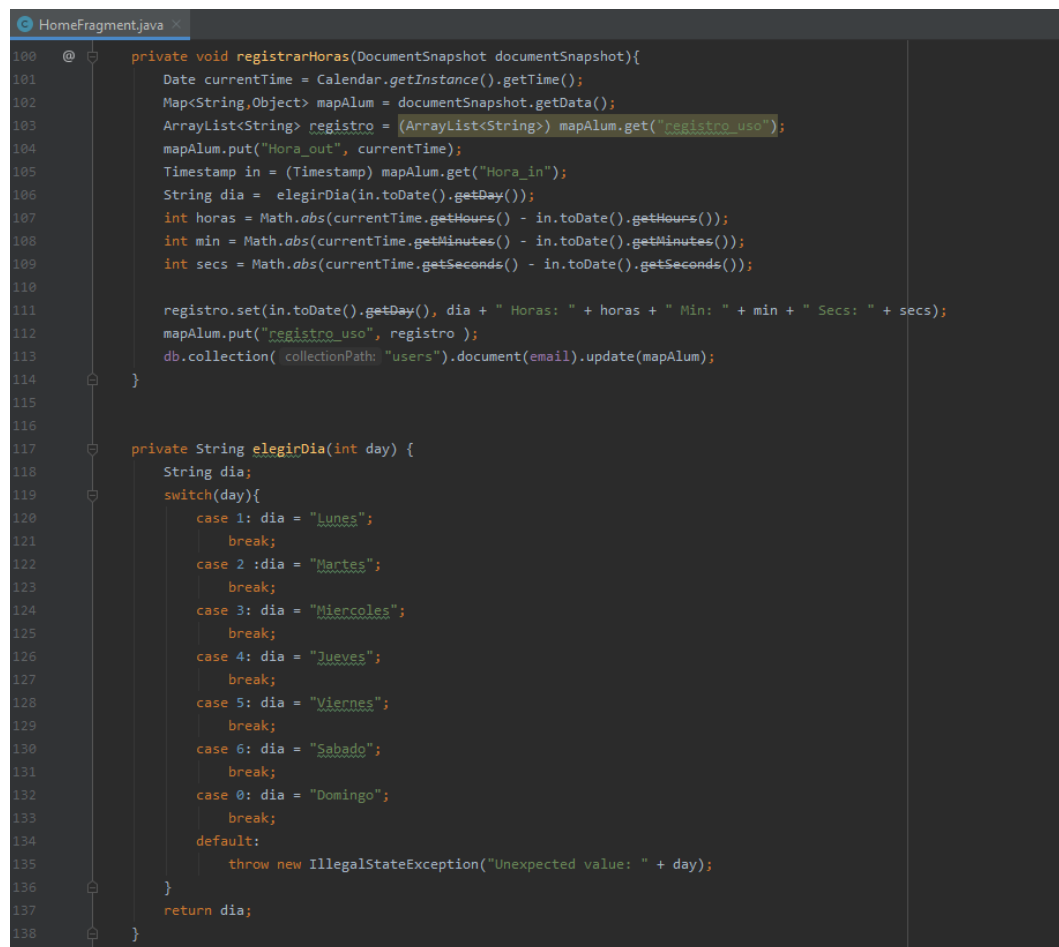
Consultar la actividad de un alumno en concreto, funcionalidad alojada en el fragmento AlumnoFragment.java (Figura 3.21). Para ello se realiza una consulta en Firestore, recogiendo la referencia del documento de Firestore con la id del alumno y mostrando sus datos (Figura 3.22).

Consultar la actividad de los alumnos de una clase (Figura 3.23), operación que se encuentra en el fragmento CursosFragment.java. Introduciendo el campo clase se hace una consulta en Firestore que devuelve los alumnos de esa clase, mostrándolos en rojo o en verde dependiendo de un límite de uso de horas determinado por el profesor.



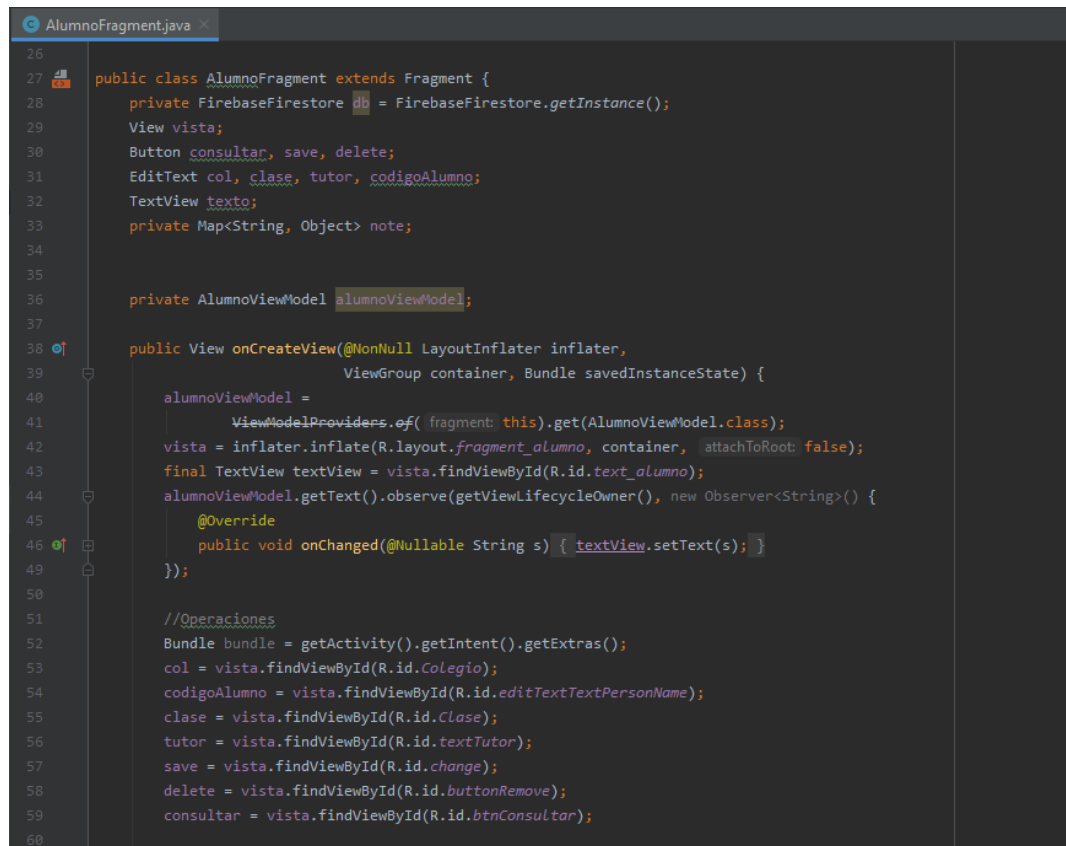
```
59 Bundle bundle = getActivity().getIntent().getExtras();
60 String email = bundle.getString( key: "email");
61 this.email = email;
62 String rol = bundle.getString( key: "rol");
63 //Cerrar sesion
64
65 logout = vista.findViewById(R.id.logout);
66
67 logout.setOnClickListener(new View.OnClickListener() {
68     @Override
69     public void onClick(View v) {
70
71         db.collection( collectionPath: "users").document(email).get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
72             @Override
73             public void onSuccess(DocumentSnapshot documentSnapshot) {
74
75                 if(documentSnapshot.getData().get("rol").toString().equalsIgnoreCase( anotherString: "alumno")) {
76                     registrarHoras(documentSnapshot);
77                 }
78                 FirebaseAuth.getInstance().signOut();
79                 getActivity().onBackPressed();
80             }
81         });
82     }
83 });
84
85 //Ir al navegador
86 browser = vista.findViewById(R.id.browser);
87
88 browser.setOnClickListener(new View.OnClickListener() {
89     @Override
90     public void onClick(View v) {
91         Intent browser = new Intent(getActivity().getApplication(), WebList.class);
92         browser.putExtra( name: "email", email);
93         startActivity(browser);
94     }
95 });
96
97 return vista;
98 }
```

Figura 3.19: Contenido del archivo HomeFragment.java



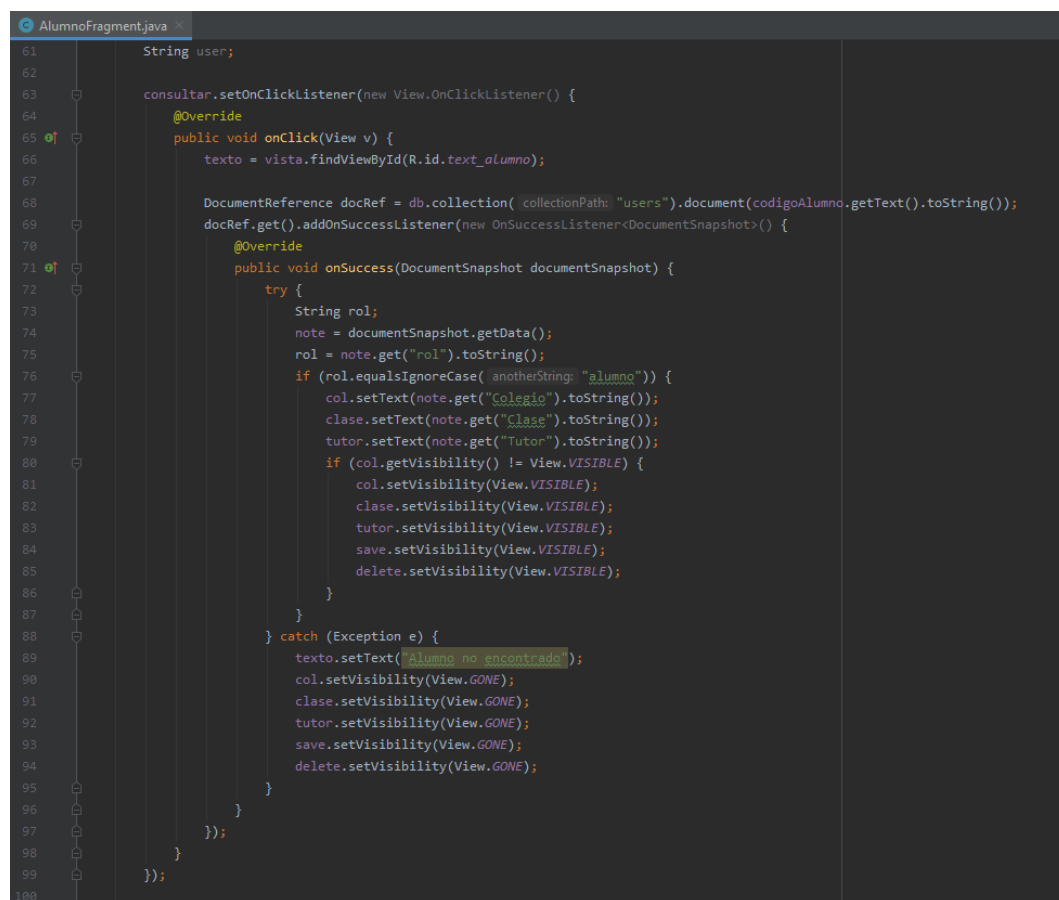
```
100 @ private void registrarHoras(DocumentSnapshot documentSnapshot){
101     Date currentTime = Calendar.getInstance().getTime();
102     Map<String,Object> mapAlum = documentSnapshot.getData();
103     ArrayList<String> registro = (ArrayList<String>) mapAlum.get("registro_uso");
104     mapAlum.put("Hora_out", currentTime);
105     Timestamp in = (Timestamp) mapAlum.get("Hora_in");
106     String dia = elegirDia(in.toDate().getDay());
107     int horas = Math.abs(currentTime.getHours() - in.toDate().getHours());
108     int min = Math.abs(currentTime.getMinutes() - in.toDate().getMinutes());
109     int secs = Math.abs(currentTime.getSeconds() - in.toDate().getSeconds());
110
111     registro.set(in.toDate().getDay(), dia + " Horas: " + horas + " Min: " + min + " Secs: " + secs);
112     mapAlum.put("registro_uso", registro);
113     db.collection( collectionPath: "users").document(email).update(mapAlum);
114 }
115
116
117 private String elegirDia(int day) {
118     String dia;
119     switch(day){
120         case 1: dia = "Lunes";
121             break;
122         case 2 :dia = "Martes";
123             break;
124         case 3: dia = "Miercoles";
125             break;
126         case 4: dia = "Jueves";
127             break;
128         case 5: dia = "Viernes";
129             break;
130         case 6: dia = "Sabado";
131             break;
132         case 0: dia = "Domingo";
133             break;
134         default:
135             throw new IllegalStateException("Unexpected value: " + day);
136     }
137     return dia;
138 }
```

Figura 3.20: Lógica del registro de uso



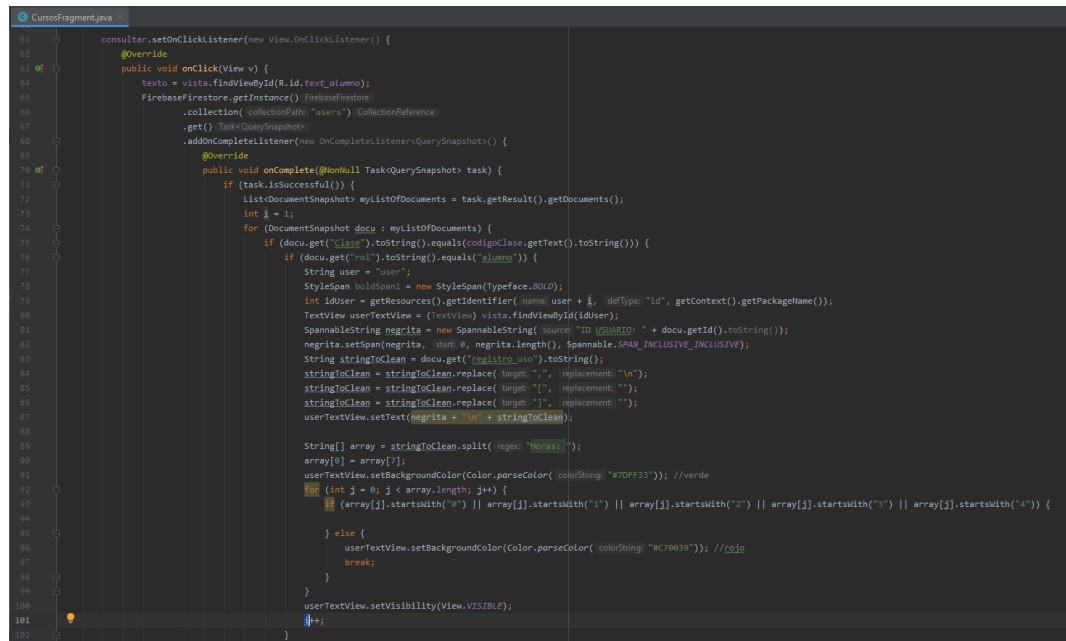
```
26
27 public class AlumnoFragment extends Fragment {
28     private FirebaseFirestore db = FirebaseFirestore.getInstance();
29     View vista;
30     Button consultar, save, delete;
31     EditText col, clase, tutor, codigoAlumno;
32     TextView texto;
33     private Map<String, Object> note;
34
35
36     private AlumnoViewModel alumnoViewModel;
37
38     public View onCreateView(@NonNull LayoutInflater inflater,
39                             ViewGroup container, Bundle savedInstanceState) {
40         alumnoViewModel =
41             ViewModelProviders.of(fragment: this).get(AlumnoViewModel.class);
42         vista = inflater.inflate(R.layout.fragment_alumno, container, attachToRoot: false);
43         final TextView textView = vista.findViewById(R.id.text_alumno);
44         alumnoViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {
45             @Override
46             public void onChanged(@Nullable String s) { textView.setText(s); }
47         });
48
49         //Operaciones
50         Bundle bundle = getActivity().getIntent().getExtras();
51         col = vista.findViewById(R.id.Colegio);
52         codigoAlumno = vista.findViewById(R.id.editTextTextPersonName);
53         clase = vista.findViewById(R.id.Clase);
54         tutor = vista.findViewById(R.id.textTutor);
55         save = vista.findViewById(R.id.change);
56         delete = vista.findViewById(R.id.buttonRemove);
57         consultar = vista.findViewById(R.id.btnConsultar);
58
59
60 }
```

Figura 3.21: Contenido del archivo AlumnoFragment.java



```
61 String user;
62
63 consultar.setOnClickListener(new View.OnClickListener() {
64     @Override
65     public void onClick(View v) {
66         texto = vista.findViewById(R.id.text_alumno);
67
68         DocumentReference docRef = db.collection("users").document(codigoAlumno.getText().toString());
69         docRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
70             @Override
71             public void onSuccess(DocumentSnapshot documentSnapshot) {
72                 try {
73                     String rol;
74                     note = documentSnapshot.getData();
75                     rol = note.get("rol").toString();
76                     if (rol.equalsIgnoreCase("alumno")) {
77                         col.setText(note.get("Colegio").toString());
78                         clase.setText(note.get("Clase").toString());
79                         tutor.setText(note.get("Tutor").toString());
80                         if (col.getVisibility() != View.VISIBLE) {
81                             col.setVisibility(View.VISIBLE);
82                             clase.setVisibility(View.VISIBLE);
83                             tutor.setVisibility(View.VISIBLE);
84                             save.setVisibility(View.VISIBLE);
85                             delete.setVisibility(View.VISIBLE);
86                         }
87                     }
88                 } catch (Exception e) {
89                     texto.setText("Alumno no encontrado");
90                     col.setVisibility(View.GONE);
91                     clase.setVisibility(View.GONE);
92                     tutor.setVisibility(View.GONE);
93                     save.setVisibility(View.GONE);
94                     delete.setVisibility(View.GONE);
95                 }
96             }
97         });
98     }
99 });
100
```

Figura 3.22: Lógica de consulta de un alumno



```

61  consultar.setOnClickListener(new View.OnClickListener() {
62      @Override
63      public void onClick(View v) {
64          texto = vista.findViewById(R.id.text_alumno);
65          FirebaseFirestore.getInstance().Firestore
66              .collection(collectionPath, "users") CollectionReference
67              .get() Task<QuerySnapshot>
68              .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
69                  @Override
70                  public void onComplete(@NonNull Task<QuerySnapshot> task) {
71                      if (task.isSuccessful()) {
72                          List<DocumentSnapshot> myListOfDocuments = task.getResult().getDocuments();
73                          int i = 1;
74                          for (DocumentSnapshot docu : myListOfDocuments) {
75                              if (docu.get("Clase").toString().equals(codigoClase.getText().toString())) {
76                                  if (docu.get("rol").toString().equals("alumno")) {
77                                      String user = "user";
78                                      StyleSpan boldSpan1 = new StyleSpan(Typeface.BOLD);
79                                      int idUser = getResources().getIdentifier( name: user + i, ddType: "id", getContext().getPackageName());
80                                      TextView userTextView = (TextView) vista.findViewById(idUser);
81                                      SpannableString negrita = new SpannableString( source: "ID USUARIO: " + docu.getId().toString());
82                                      negrita.setSpan(negrita, 0, negrita.length(), Spannable.SPAN_INCLUSIVE_INCLUSIVE);
83                                      String stringToClean = docu.get("registro-uso").toString();
84                                      stringToClean = stringToClean.replace( target: ",", replacement: "v");
85                                      stringToClean = stringToClean.replace( target: ":", replacement: "");
86                                      stringToClean = stringToClean.replace( target: "]", replacement: "");
87                                      userTextView.setText(negrita + "\n" + stringToClean);
88
89                                      String[] array = stringToClean.split( regex: "horas:");
90                                      array[0] = array[1];
91                                      userTextView.setBackgroundColor( Color.parseColor( colorHex: "#70FF33")); //verde
92                                      for (int j = 0; j < array.length; j++) {
93                                          if (array[j].startsWith("0") || array[j].startsWith("1") || array[j].startsWith("2") || array[j].startsWith("3") || array[j].startsWith("4")) {
94                                              } else {
95                                                  userTextView.setBackgroundColor( Color.parseColor( colorString: "#C70039")); //rojo
96                                                  break;
97                                              }
98                                          }
99                                      }
100                                      userTextView.setVisibility(View.VISIBLE);
101                                  }
102          }

```

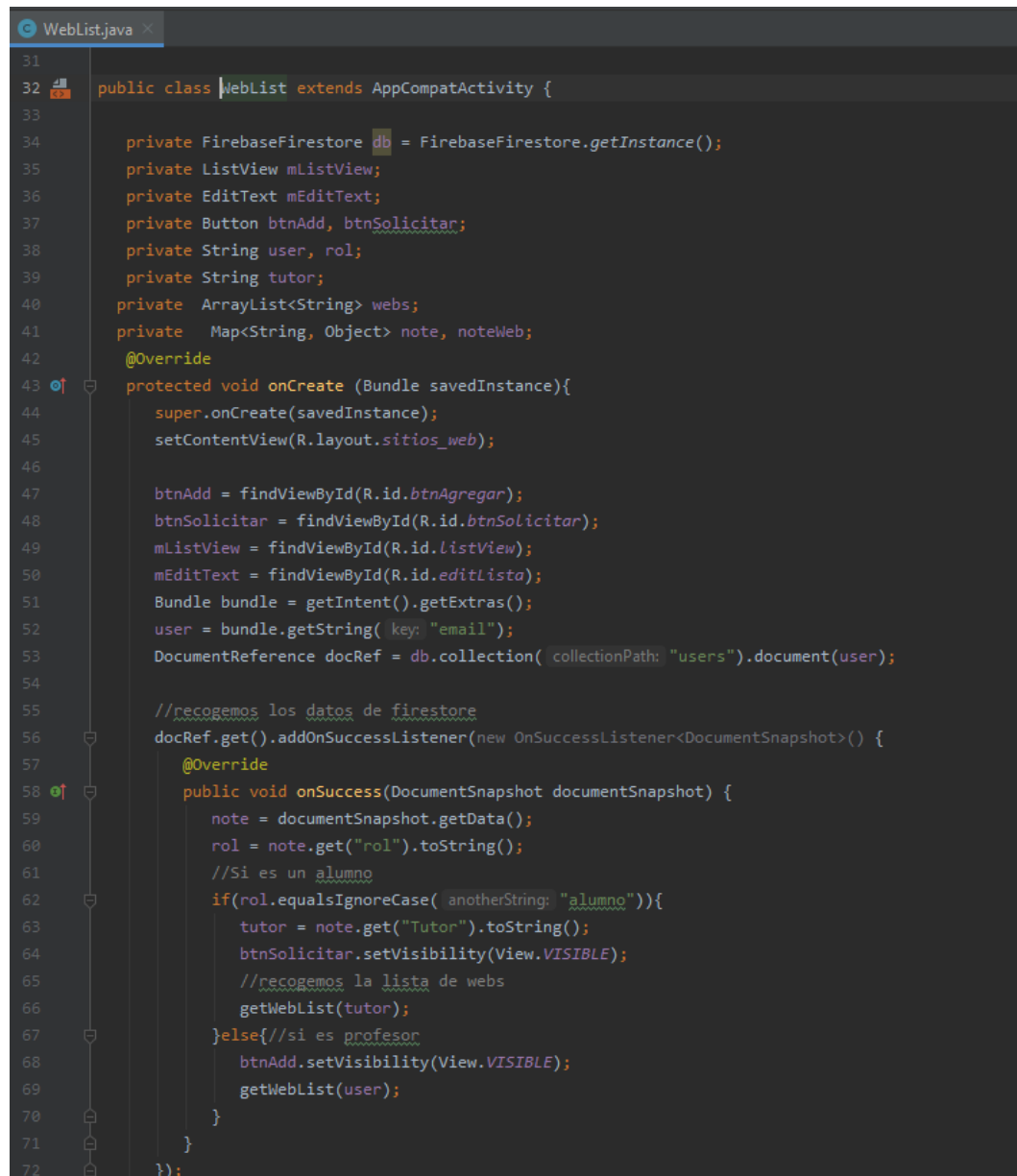
Figura 3.23: Lógica de consulta de un curso

3.4.7. Listado de webs

En este apartado se recoge la lista de webs que tiene el usuario asignado. La lógica está en la clase WebList.java (Figura 3.24). En el caso del profesor, sólo se recoge el array que tiene en sus propiedades de Firestore; en el lugar del alumno, se hace una consulta recogiendo la identificación del tutor y mostrando la lista de sitios web asignada.

El método `getWebList` (Figura 3.25) es el encargado de mostrar la lista de webs, primero recogiendo los datos de Firestore y luego mostrándola mediante un `ListView`, que es un adaptador de datos que los muestra verticalmente y con scroll incorporado. En él se ha implementado un evento de entrada llamado `onItemClickListener` (Figura 3.26), encargado de que si se hace un click en la posición en la que está la dirección web, llame a la actividad `WebBrowser.java` (Figura 3.27), que será la encargada de cargar dicha url y mostrarla dentro de la aplicación.

La carga de una url se hace gracias a un tipo de vista llamada `WebView`, que permite la muestra del contenido web como parte de la actividad del layout. Resulta muy útil para incrementar el control de la interfaz y de esta manera se evita que cualquier usuario acceda al navegador del sistema por defecto.



```
31
32 public class WebList extends AppCompatActivity {
33
34     private FirebaseFirestore db = FirebaseFirestore.getInstance();
35     private ListView mListView;
36     private EditText mEditText;
37     private Button btnAdd, btnSolicitar;
38     private String user, rol;
39     private String tutor;
40     private ArrayList<String> webs;
41     private Map<String, Object> note, noteWeb;
42     @Override
43     protected void onCreate (Bundle savedInstanceState){
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.sitios_web);
46
47         btnAdd = findViewById(R.id.btnAgrega);
48         btnSolicitar = findViewById(R.id.btnSolicitar);
49         mListView = findViewById(R.id.listView);
50         mEditText = findViewById(R.id.editLista);
51         Bundle bundle = getIntent().getExtras();
52         user = bundle.getString( key: "email");
53         DocumentReference docRef = db.collection( collectionPath: "users").document(user);
54
55         //recogemos los datos de firestore
56         docRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
57             @Override
58             public void onSuccess(DocumentSnapshot documentSnapshot) {
59                 note = documentSnapshot.getData();
60                 rol = note.get("rol").toString();
61                 //Si es un alumno
62                 if(rol.equalsIgnoreCase( anotherString: "alumno")){
63                     tutor = note.get("Tutor").toString();
64                     btnSolicitar.setVisibility(View.VISIBLE);
65                     //recogemos la lista de webs
66                     getWebList(tutor);
67                 }else{//si es profesor
68                     btnAdd.setVisibility(View.VISIBLE);
69                     getWebList(user);
70                 }
71             }
72         });
73     }
74 }
```

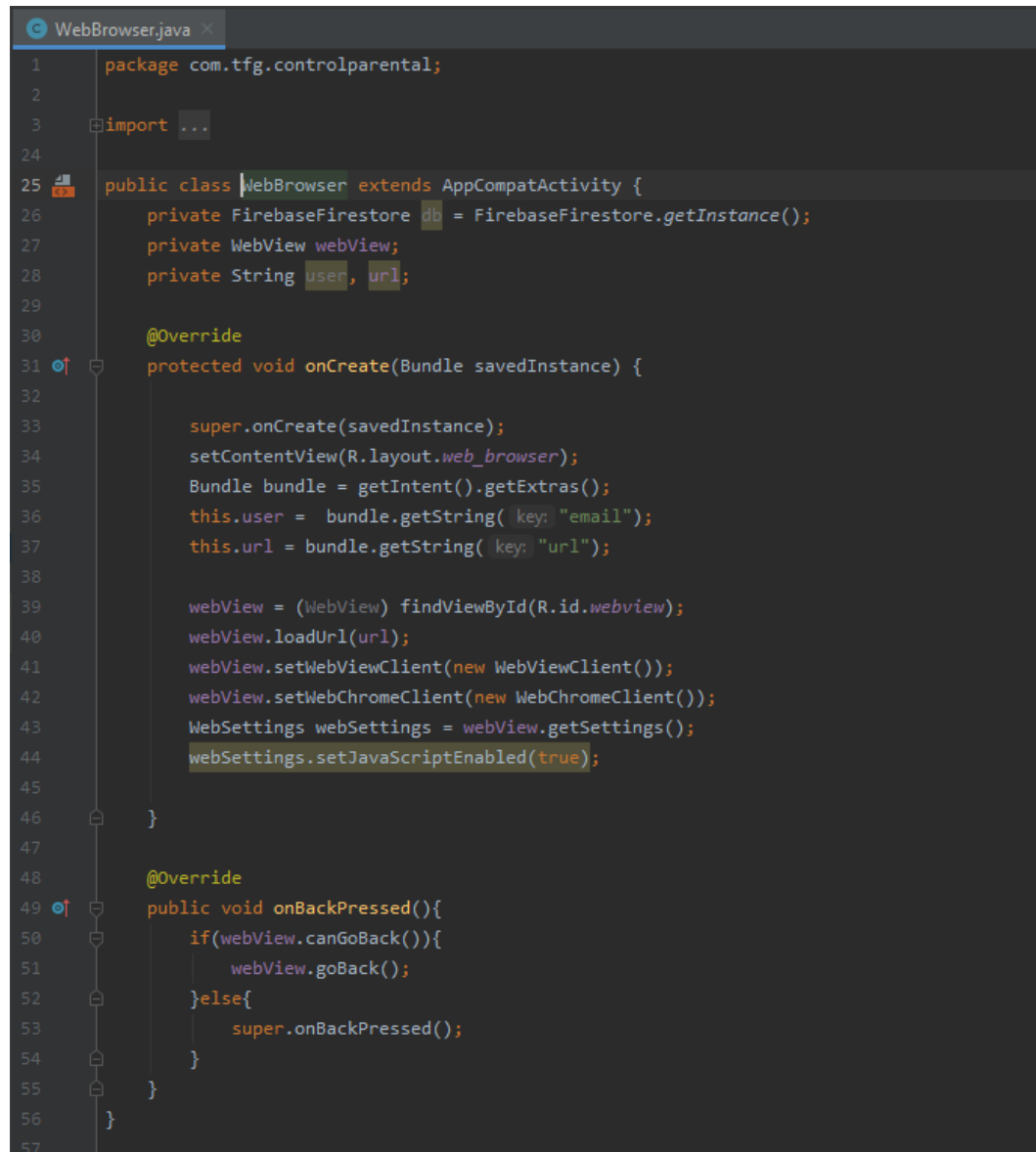
Figura 3.24: Contenido de la Clase WebList

```
private void getWebList(String tutor) {  
    db.collection( collectionPath: "users").document(tutor).get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {  
        @Override  
        public void onSuccess(DocumentSnapshot documentSnapshot) {  
            noteWeb = documentSnapshot.getData();  
            webs = (ArrayList<String>) noteWeb.get("searches");  
            ArrayAdapter<String> adapter = new ArrayAdapter<>( context: WebList.this, android.R.layout.simple_list_item_1, webs);  
            mListView.setAdapter(adapter);  
        }  
    });  
}
```

Figura 3.25: Método getWebList

```
mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        Intent web = new Intent( packageContext: WebList.this, WebBrowser.class);  
        String url = (String) (mListView.getItemAtPosition(position));  
        web.putExtra( name: "url", url);  
        startActivity(web);  
    }  
}); //end_OnItemClick_mListView
```

Figura 3.26: Evento de los elementos de la lista



```
1 package com.tfg.controlparental;
2
3 import ...
4
25 public class WebBrowser extends AppCompatActivity {
26     private FirebaseFirestore db = FirebaseFirestore.getInstance();
27     private WebView webView;
28     private String user, url;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.web_browser);
35         Bundle bundle = getIntent().getExtras();
36         this.user = bundle.getString( key, "email");
37         this.url = bundle.getString( key, "url");
38
39         webView = (WebView) findViewById(R.id.webview);
40         webView.loadUrl(url);
41         webView.setWebViewClient(new WebViewClient());
42         webView.setWebChromeClient(new WebChromeClient());
43         WebSettings webSettings = webView.getSettings();
44         webSettings.setJavaScriptEnabled(true);
45
46     }
47
48     @Override
49     public void onBackPressed(){
50         if(webView.canGoBack()){
51             webView.goBack();
52         }else{
53             super.onBackPressed();
54         }
55     }
56 }
57
```

Figura 3.27: Clase WebBrowser

Capítulo 4

Resultados obtenidos

En este capítulo se muestran los resultados del desarrollo de la aplicación. Desde la autenticación, navegación entre pantallas, hasta las funcionalidades incorporadas para el control escolar.

4.1. Autenticación

Cuando se ejecuta la aplicación, lo primero que se encuentra es el contenido de la Figura 4.1, correspondiente a la pantalla de autenticación. Se ven dos campos, uno para introducir el correo electrónico y otro para la contraseña, seguido de dos botones, uno de registro y otro de acceso a la aplicación. En la Figura 4.2 se muestra la pantalla que aparece tras pinchar en botón de registro, el contenido que se observa son 4 campos correspondientes al e-mail, clase, colegio y contraseña. Si hay algún campo vacío salta un mensaje de error.

4.2. Inicio y menú de opciones

Si no existe la cuenta introducida en el registro y no falta ningún campo, se insertará el nuevo usuario en Firestore y se añadirá a la lista de Firebase Authentication. Inmediatamente se redirigirá a la pantalla de home, como se muestra en la Figura 4.3, donde aparece un botón de cerrar sesión, otro que lleva a la lista de webs permitidas, además del menú de opciones (Figura 4.4), al cual sólo puede acceder el profesor. Para acceder a este menú de opciones arrastramos el ratón para abrir el navegador de opciones o pinchamos en el menú de hamburguesa y observamos varios espacios, uno para consultar alumnos, otro para consultar una clase concreta y otro para gestionar opciones.

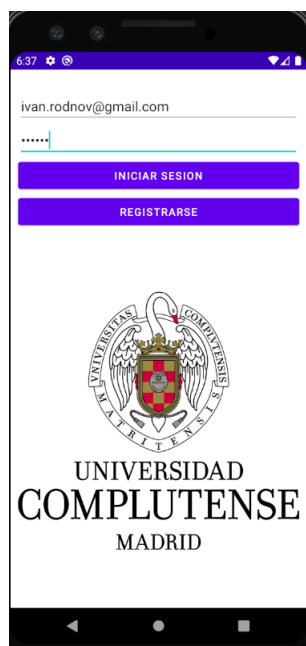


Figura 4.1: Pantalla de autenticación

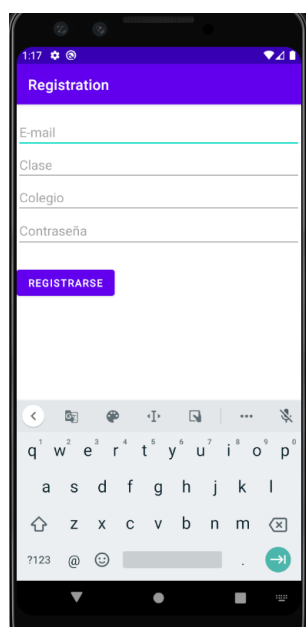


Figura 4.2: Pantalla de registro

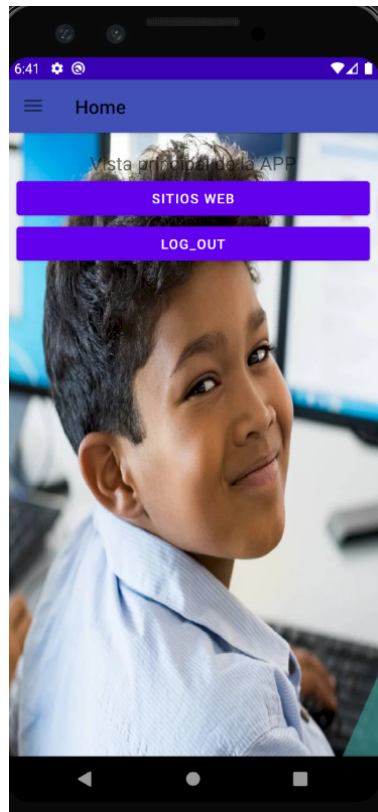


Figura 4.3: Pantalla de Inicio

Dentro de la opción Alumno vemos un campo, donde se debe introducir el correo de alumno que se quiere consultar, una vez introducido pinchamos en el botón Consultar se ve el contenido mostrado en la Figura 4.5, donde se ven los valores asignados a ese alumno seguido de dos botones, uno para editar los campos y otro para eliminar el usuario. En la opción de Cursos, vemos un campo donde se introduce una clase con el formato número-letra, por ejemplo, 3A, seguido de un botón para consultar los alumnos de esa clase. Si ejecutamos una consulta, vemos el contenido de la Figura 4.6, una lista de alumnos donde se ve el registro de las horas de cada uno de ellos, aquellos que no se han pasado ningún día del límite de horas de uso, determinado por el profesor, aparecen sobre un cuadro verde, y en rojo los alumnos que al menos algún día han estado más tiempo del asignado.

4.3. Lista de webs

Esta funcionalidad es distinta según el tipo de usuario, como se observa en la Figura 4.7, si el usuario tiene el rol de profesor, aparecerá un botón para agregar una url a la lista, así como un campo de texto para que escriba

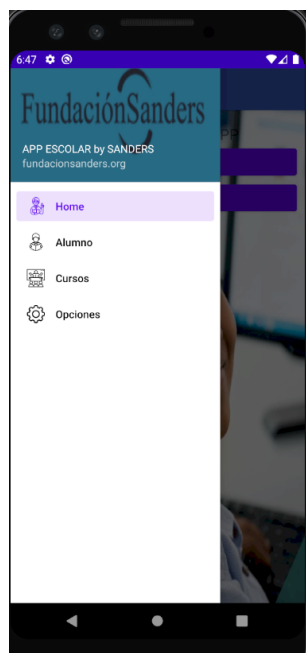


Figura 4.4: Menú de opciones

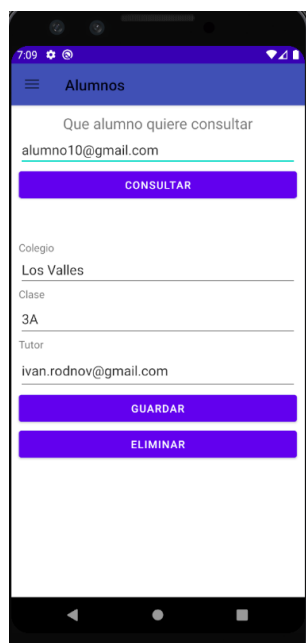


Figura 4.5: Alumno Consultado

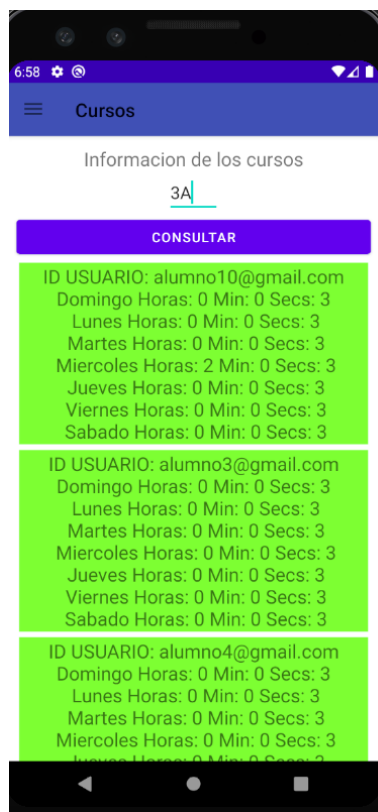


Figura 4.6: Menú de opciones



Figura 4.7: Listado de webs, versión del profesor

el sitio web que desee, debajo del botón de agregar, aparece una lista de los sitios web que ha agregado el profesor. Por el contrario, si el usuario tiene el rol de alumno se mostrará el contenido que observamos en la Figura 4.8, un campo para escribir la web que quiera el usuario, un botón donde puede solicitar la agregación de un sitio web y la lista de sitios a los que puede acceder. Si se hace click en un elemento de la lista de webs, se mostrará el contenido de la web dentro de la propia aplicación, como se muestra en la Figura 4.9. De esta forma se evita que los alumnos accedan al navegador.

4.4. Resumen de los resultados

La aplicación tiene las funcionalidades previstas y hemos visto cómo hemos llegado a estos resultados partiendo desde cero, reflejándolos en una serie de pantallas y explicando en cada apartado su función y cómo se ven representados los escenarios de uso, además de ver más a fondo las opciones que tiene cada tipo de usuario y cómo debe responder el sistema.



Figura 4.8: Listado de webs, versión del alumno

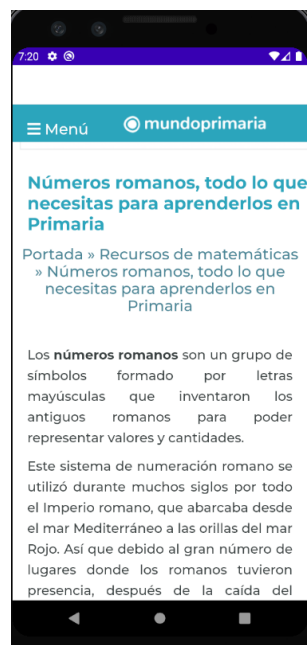


Figura 4.9: Contenido de la web cargada

Capítulo 5

Conclusiones y Trabajo Futuro

Este capítulo cierra la memoria con las conclusiones obtenidas tras la realización de este trabajo de fin de grado, así como presentando algunas de las dificultades que han ido apareciendo a lo largo de su desarrollo, y las soluciones planteadas. Por último se propondrán trabajos futuros que pueden ampliar el proyecto, para mejorarlo y que sea más completo.

5.1. Conclusiones

Se ha conseguido desarrollar una aplicación para el sistema operativo Android, con el fin de proporcionar una solución a la problemática planteada por la Fundación Sanders sobre el control de uso de los dispositivos digitales que han repartido a familias desfavorecidas.

La motivación que nos ha llevado a realizar este proyecto, surgió de una reunión con Alma Orozco, representante de la Fundación Sanders, que nos explicó la problemática que tenían con el uso de estas tabletas por menores de algunas familias dentro el ámbito de la educación. Una vez entendimos el problema, ayudar a esas familias era razón más que suficiente para ponerse a trabajar.

En una primera fase comenzamos a investigar las distintas aplicaciones que existían en el mercado. Descubrimos que eran todas muy parecidas, y que apuntaban siempre al control de otras aplicaciones y a tener un registro de actividades, además de tener dos modos de uso, una versión para padres o tutores y otra para los menores, teniendo distintas vistas y opciones para cada uno.

Tras este estudio y análisis, decidimos comenzar el desarrollo de una aplicación Android, ya que es el sistema operativo que más porcentaje de usuarios

tiene, para así llegar al mayor número de personas posible. Para el desarrollo de la herramienta, utilizando los conocimientos y aplicaciones descritas en los capítulos de la memoria, intentamos diseñarla de forma que el menor no se sienta controlado en todo momento, pero que a la vez se asegurara un uso únicamente académico de los dispositivos digitales puestos a su servicio por la Fundación Sanders con ese fin.

El proyecto ha conseguido que hayamos desarrollado una aplicación que cumple con los objetivos principales propuestos: limitar el acceso a un listado de sitios web determinados por un profesor, que sólo se navegue por páginas con contenido exclusivamente educativo, y limitar el tiempo de uso. Además los datos de las webs que son académicas se cargan en la propia aplicación, sin necesidad de ejecutar el navegador del dispositivo, para así evitar que entren en sitios no deseados. También hemos logrado que el usuario profesor pueda monitorizar el tiempo de uso que hacen los alumnos de esta aplicación, consultando una clase concreta, de forma que si detecta una acción sospechosa, se encargue él mismo de tomar las medidas correspondientes.

Hemos partido de cero para diseñar una aplicación que puede ser la base de la solución que busca la Fundación Sanders. Para hacerla más completa, se podría ampliar y mejorar, aumentando los casos de uso y estudiando cómo hacer una carga de datos masiva, para que sea lo más eficiente posible en situaciones reales, de forma que resulte útil y robusta.

5.2. Trabajo Futuro

A la vista de los resultados, nos planteamos como posibles trabajos futuros los siguientes:

- Implementar la carga de usuarios de forma que se detecte y se evite que un usuario mal intencionado intente crear una cuenta con el rol de profesor. De esta manera se evita desarrollar dos versiones de la aplicación, como hacen muchas de las aplicaciones ya existentes.
- Desarrollar una solución para que el profesor reciba un informe semanal con el registro de uso que han tenido los alumnos de una clase de estos dispositivos digitales.
- Implementar un sistema de comunicación entre alumno y profesor, como una bandeja de entrada de un correo electrónico; así los alumnos pueden preguntar sus dudas o solicitar sitios web que les pueda interesar.

- Crear una base de datos mejor optimizada y con una estructura más avanzada para la carga de datos de varios colegios a la vez.
- Hacer una plantilla de diseño para que la aplicación sea más atractiva visualmente para todos los usuarios, y que resulte fácil y cómoda de usar.
- Desarrollar una secuencia de tests para asegurar el correcto funcionamiento de la aplicación.

This chapter closes the report with the conclusions obtained after completing this final degree project, as well as presenting some of the difficulties that have appeared throughout its development, and the solutions proposed. Finally, future works will be proposed that can expand the project, to improve it and make it more complete.

5.3. Conclusions

An application for the Android operating system has been developed in order to provide a solution to the problem raised by the Sanders Foundation regarding the control of the use of digital devices that have been distributed to disadvantaged families.

The motivation that has led us to carry out this project arose from a meeting with Alma Orozco, representative of the Sanders Foundation, who explained to us the problems they had with the use of these tablets by minors from some families within the field of education. Once we understood the problem, helping those families was more than enough reason to get to work.

In a first phase we began to investigate the different applications that existed in the market. We discovered that they were all very similar, and that they always aimed to control other applications and have an activity log, in addition to having two modes of use, a version for parents or guardians and another for minors, having different views and options for each one.

After this study and analysis, we decided to start the development of an Android application, since it is the operating system with the highest percentage of users, in order to reach as many people as possible. For the development of the tool, using the knowledge and applications described in the chapters of the report, we try to design it in such a way that the minor does not feel controlled at all times, but at the same time ensures a solely academic use of digital devices placed at your service by the Sanders Foundation for that purpose.

The project has ensured that we have developed an application that meets the main objectives proposed: limit access to a list of websites determined by a teacher, only browse pages with exclusively educational content, and limit the time of use. In addition, the data from the academic websites is loaded into the application itself, without the need to run the device's browser, in order to prevent them from entering unwanted sites. We have also achieved that the teacher user can monitor the time of use that students make of this application, consulting a specific class, so that if he detects a suspicious action, he will take the corresponding measures himself.

We have started from scratch to design an application that can be the basis of the solution that the Sanders Foundation is looking for. To make it more complete, it could be expanded and improved, increasing the use cases and studying how to do a massive data load, so that it is as efficient as possible in real situations, in a way that is useful and robust.

5.4. Future work

In view of the results, we consider the following as possible future works:

- Implement user loading in such a way that a malicious user is detected and prevented from attempting to create an account with the teacher role. In this way, you avoid developing two versions of the application, as many of the existing applications do.
- Develop a solution for the teacher to receive a weekly report with the record of use that students in a class have had of these digital devices.
- Implement a communication system between student and teacher, such as an email inbox; so students can ask their questions or request websites that may interest them.
- Create a better optimized database with a more advanced structure for loading data from several schools at the same time.
- Make a design template so that the application is more visually attractive for all users, and that it is easy and comfortable to use.
- Develop a sequence of tests to ensure the correct operation of the application.

Capítulo 6

Aportación de los alumnos

Cada alumno ha desarrollado distintas tareas a lo largo del trabajo. Aquí se indica la aportación de cada uno:

6.1. Aportación del alumno Iván Rodríguez Novoa

Este alumno ha participado en los siguientes desarrollos del trabajo fin de grado:

Investigación de aplicaciones

Durante las primeras reuniones, nos recomendaron investigar aplicaciones que ya existieran en el mercado con objetivos similares a la que íbamos a desarrollar, con el fin de tener una idea de lo que debería tener nuestra aplicación. Primero busqué en Google Académico con el objetivo de encontrar referencias sobre el control parental y escolar, y estudios sobre la situación de los menores con respecto al uso de la red. Encontré muchas similitudes entre los problemas que encontraban tanto padres como profesores, como que no pueden asegurar un uso correcto de Internet por parte de sus hijos o alumnos. Decidimos investigar aplicaciones que ya existían que proponían soluciones para monitorizar otros dispositivos. Por mi parte estudié las aplicaciones Qustodio y Family Link; además instalé sus versiones demo en varios dispositivos para experimentar de forma práctica su funcionamiento. Estas aplicaciones están documentadas en la memoria en los puntos 2.1.1 y 2.1.2.

Comparación de aplicaciones

Una vez acabada la investigación de las aplicaciones que existen, me encargé de hacer una tabla comparativa, recogiendo las características comunes que compartían y comparandolas entre sí con el fin de ver lo que debíamos aportar en nuestra aplicación, así como nuevas funcionalidades que vieramos necesarias para poder diferenciarse de las demás.

Planteamiento de escenarios

Tras haber hecho la investigación del estado del arte, se propuso realizar una estructura para la base de datos así como casos de uso. Me ocupé de este apartado, pensando en una estructura simple y concreta para su fácil implementación, además de proponer algún caso de uso como funcionalidades de la aplicación. Para ello utilicé una herramienta llamada Lucidchart, un espacio web donde se pueden crear diagramas y visualizar datos. Estos diagramas se pueden ver en la sección 3.2.

Investigación de Firebase

Una vez empezamos el desarrollo de la aplicación móvil, había que pensar en la forma de guardar datos dentro de la nube, ya que queríamos usar un espacio donde tener almacenada la información de los usuarios de la aplicación. Como decidimos desarrollarla en Android, nos recomendaron investigar Firebase, una plataforma creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de una forma rápida. Indagué más a fondo dentro del propio servicio, agregando tanto usuarios como cuentas ficticias, y comprobé que su uso era sencillo. Además pude configurarlo dentro del proyecto de la aplicación sin mucha dificultad.

Autenticación

Ya comenzado el desarrollo de la aplicación, lo primero que había que hacer era implementar el proceso de autenticación, y que fuera compatible con el servicio de Firebase. Decidí hacerlo de una manera estándar, introduciendo un correo electrónico y la contraseña asociada. Con respecto al registro, lo desarrollé con el fin de comprobar que el servicio Firebase funcionaba, como se explica en las conclusiones en el punto 5.1. Estaba claro que no debería haber un apartado de registro, ya que un usuario podría hacer un mal uso de ello.

Listado de webs

Cuando llegó la hora de abordar el problema principal, investigué cómo poder capturar una llamada a un sitio web, y si existía una manera de recoger la url y bloquearla o aceptarla. Había posibles opciones, como usar Volley, una biblioteca HTTP que agiliza el uso de redes en aplicaciones Android, configurando una cola de peticiones y accediendo a la caché de opciones. Sin embargo, era un procedimiento complicado y que no conseguía que funcionara bien del todo, así que al comentarlo en las reuniones, me recomendaron directamente hacer una lista de sitios web, la cual únicamente un profesor podría editar y que no se pudiera acceder al navegador del dispositivo. De esta manera se aseguraba que los menores hicieran un uso únicamente académico,

accediendo a los elementos del listado. Con el uso de WebView conseguí que el contenido web se cargara dentro de la aplicación.

Registro Horario

En una de las reuniones, ya avanzado el proyecto, nos recomendaron que para tener mayor facilidad de monitorización sobre el uso de los menores de los dispositivos, hubiera un registro diario del uso de la aplicación. Para ello intenté iniciar un contador de tiempo cada vez que se accede a la aplicación y que se para cuando un usuario cerrara sesión, pero no lo devolvía en un formato cómodo para la inserción de ese dato en Firestore. Entonces decidí hacerlo con un tipo de datos que admitía Firestore dentro de sus documentos, que es un timestamp; de esta manera calculaba el tiempo, además del día, que pasaba el usuario desde que iniciaba hasta que cerraba la sesión.

Desarrollo de la memoria

A la hora de documentar, nos aconsejaron utilizar LaTeX como entorno de composición de textos. Desarrollamos un índice de objetivos desde el comienzo del trabajo, por lo que no resultó difícil estructurar la memoria, además de aprender a utilizar una nueva herramienta. Me encargué de la redacción de muchos de los puntos de la documentación, así como de buscar citas y conceptos en la bibliografía.

6.2. Aportación del alumno Carlos Sanz Tomero

Este alumno ha participado en los siguientes desarrollos del trabajo fin de grado:

Investigación de aplicaciones

Como se ha comentado, nos recomendaron investigar sobre herramientas comerciales y software similar a lo que queríamos hacer, que ya existiera, para hacernos una idea de las funcionalidades a implementar. Yo me centré en Kroha, donde me di de alta como usuario tutor y usuario alumno para tomar capturas e investigar la aplicación. A parte de Kroha, investigué otras herramientas de control parental para tener una recogida de datos más amplia y poder hacer una comparación entre todas ellas.

Menú lateral

Me encargué de la implementación del menú lateral. La navegación en la aplicación debe ser lo más cómoda e intuitiva posible, por lo que al completar la autenticación en la misma, se nos redirige a la ventana principal. Desde esta ventana podemos abrir el menú, tanto arrastrando desde el lateral como tocando en el icono del menú. Una vez desplegado podremos acceder a las demás utilidades presentes en la aplicación, gracias al elemento navigation. Posteriormente coloqué el logo de la Fundación Sanders.

Herramienta de navegación

Para poder navegar entre los fragments de nuestra aplicación, a través del menú anteriormente mencionado, fue necesario crear un elemento navigator, desde el cual se pudiese ir pasando de un fragment a otro. Esta fue otra de las tareas de las que me encargué.

Iconos open source

Para hacer más clara la navegación entre los menús, consideré necesario añadir unos iconos que fuesen los más intuitivos posibles al menú, de tal manera que de un simple vistazo se pudiese visualizar cada opción. La elección de los mismos se hizo entre una librería de iconos open source, para no tener ningún problema de derechos.

Alumno

Una vez están registrados los alumnos, el profesor debería de poder consultar todos los datos del alumno, así como modificarlos o eliminar al propio alumno del sistema. Diseñé e implementé la ventana que permite hacer esto, así como consultar a través del email del alumno todos los datos almacenados

y poder editarlos o eliminarlos. Todo esto se hace a través de la gestión de la base de datos que usamos, Firebase.

Cursos

La principal utilidad de la aplicación es el control de las horas que los alumnos pasan haciendo uso de su dispositivo digital, por lo que era indispensable poder consultar un curso y ver, de una manera rápida y sencilla, qué alumnos estaban haciendo un buen uso de su dispositivo y cuáles no. Se decidió hacerlo a través de un código de colores muy simple, rojo para el alumno que ha superado el número de horas de uso permitido y verde para el que lo ha cumplido. En esta pestaña se muestra esto, junto con su email y el registro horario. Todo esto se ha llevado a cabo a través de la consulta de datos a Firebase y su posterior tratamiento.

Opciones

También se nos planteaba otro problema, ¿Deberían alumnos de diferentes cursos y edades, tener el mismo número de horas de uso? Esta pestaña surge de esa necesidad. Desde aquí el profesor puede modificar el número de horas que permite a sus alumnos usar el dispositivo sin que salte la alerta, pudiendo modificarlo en cualquier momento.

Carga de datos

Una vez acabadas las funcionalidades necesarias, me encargué de hacer una carga de datos de usuarios en Firestore para poder simular un proceso más cercano a la realidad y poder comprobar que la aplicación respondía correctamente frente a varios elementos, sobre todo para la funcionalidad de Cursos y para la lista de webs asociada al usuario con rol de profesor.

Retoques finales

Cuando la aplicación estaba casi desarrollada en su totalidad, surgió la necesidad de aumentar la calidad gráfica de la misma, ya que cumplía la funcionalidad para la que había sido diseñada, pero no era todo lo visual que se esperaba. Así que me encargué de configurar el scroll en las pantallas en las que era necesario, y de ajustar el formato de los botones, hacer dinámicos múltiples elementos para que en función de la cantidad a mostrar se desplazasen, y añadir algunas imágenes como el logo de la Universidad Complutense en la pestaña de login.

Bibliografía

- [1] Isabel Bartau-Rojas, Ana Aierbe-Barandiaran, and Eider Oregui-González. Mediación parental del uso de internet en el alumnado de primaria: creencias, estrategias y dificultades. *Comunicar*, 26(54):71–79, 2018.
- [2] Base de datos nosql, definición, tipos, ventajas y desventajas. <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/> (visited: 2021-05-16).
- [3] Kidbox home. <https://www.kidsboxapps.es/> (visited: 2021-05-16).
- [4] Google Developers. Firebase documentation. <https://firebase.google.com/docs> (visited: 2021-05-16).
- [5] Google family link. <https://families.google.com/intl/es/familylink/> (visited: 2021-05-16).
- [6] Firebase. <https://firebase.google.com/?hl=es> (visited: 2021-05-16).
- [7] Guido Flammey, Verónica Gubbins, Francisca Morales, et al. Los centros de padres y apoderados: nuevos actores en el control de la gestión escolar. *CIDE*, 1999.
- [8] Google cloud platform. <https://console.cloud.google.com/> (visited: 2021-05-16).
- [9] Webfisy. <http://wefisy.com/?lang=es> (visited: 2021-05-16).
- [10] Kaspersky safe kids. <https://www.kaspersky.es/safe-kids> (visited: 2021-05-16).
- [11] Lucidchart. <https://www.lucidchart.com/pages/es> (visited: 2021-05-16).
- [12] maxmadrid. página de inicio de maxmadrid. <https://www.educa2.madrid.org/web/max> (visited: 2021-05-30).

- [13] Yesenia Morales Alonso et al. El control parental en la regulación del uso de las redes sociales en adolescentes: influencia de las redes sociales en las relaciones entre estos y estas. 2021.
- [14] Kroha description. https://play.google.com/store/apps/details?id=ua.com.tim_berners.parental_control&hl=es&gl=US (visited: 2021-05-16).
- [15] Profuturo. página principal de profuturo. <https://profuturo.education/> (visited: 2021-05-30).
- [16] Qustodio. <https://www.qustodio.com/es/> (visited: 2021-05-16).
- [17] Fundación Sanders. Página principal de la fundación sanders. <https://www.fundacionsanders.org/> (visited: 2021-05-30).
- [18] Víctor José Villanueva-Blasco and Sara Serrano-Bernal. Patrón de uso de internet y control parental de redes sociales como predictor de sexting en adolescentes: una perspectiva de género. *Revista de Psicología y Educación*, 14(1):16–26, 2019.